Reuse of IP and Virtual Components

Ralf Seepold

Forschungszentrum Informatik an der Universität Karlsruhe (FZI)
Haid-und-Neu-Str. 10-14
D-76131 Karlsruhe, Germany

Abstract

Reuse of IP and virtual components can support both bridging the "design gap" and realizing reuse-based System-on-Chip design. As a fundamental effort, design of common interface standards for virtual components provide features required to accommodate IP and to transfer IP in a virtual market place, i.e., introducing a methodological shift from intracompany reuse to intercompany application. IPs are described and categorized and the most common types are presented by their distinctive characteristics. The viewpoint of designers, providers and users is sketched and related to the roadmap and IP business models, in order to visualize the requirements for an IP market. From the large number of ongoing research work, two developments have been selected to show the shift to a new paradigm: An IP protection methodology that is part of a standardization initiative, and a concrete implementation of a reuse management system that supports intercompany reuse of IP.

1 Introduction

The application of reuse and reuse-oriented design techniques seem to be the key to avoid a widening of the "design gap" that is referred to chip capacity versus design capabilities. Taking into account that chip capacity increases by 58% per year, CAD tool support only increases by 21% per year [Semi97]. Design reuse as a key technology is a paradigm that can be compared to that of high-level synthesis until a few years ago.

In the moment several solutions are offered to close the gap between available IC technology and EDA tool capabilities [Seep98b]:

- Development of powerful EDA design tools
- Customization and development of (in-house) business models
- Methodologies for design and reuse of IPs
- Rapid prototyping support
- Exchange of IPs

Furthermore, design reuse and the definition of virtual components (VC) [Virt97] support companies' effort to realize a System-On-Chip (SoC) design. Since SoC design must face the requirements of today's consumer market. Currently, the IP-oriented market is in the process of being developed and installed: different sources of IP, short life-cycles and continuously low efforts for using and maintaining VCs are cornerstones of a prosperous future of this virtual market. These cornerstones meet the consumers' demands which are the prevalent driving forces behind IP: cut-down costs, increase functionality and shorten time to market.

To summarize the market-driven forces and the impact on ECAD, the first and most important aspect of motivation remains the decrease of development time. Products have to be conform to interoperability standards, otherwise, exchange of IP cannot be realized. In the future, companies focus on core competencies, while they will buy expertise in complementary areas. Powerful and flexible IP will be important to establish product derivatives and customizations.

One impact is that decisions will be made on system-level to determine cost, performance and viability. Any support of virtual prototyping will become vital and a rapid move to VC-block based construction will emerge, while in parallel, appropriate standards are requested. IP will be a product that is available not only in-house but on the market. Design methodologies must support IP and HW/SW implementation flows that meet the requirements of a design for reuse strategy.

2 Virtual Components

The term *IP* occurs very often in the discussion on reuse. Until now, no accepted definition is available that covers all aspects while avoiding an overloading description. IP is more closely related to legal terms like copyright, patents or warranty. By contrast, a virtual component is well defined [Virt97]. A virtual component is pre-implemented and designed for reuse, i.e., it is an reusable module. VCs contain IP and as an important feature, VCs can be quickly inserted into an existing design or specification, respectively. But the application of a VC does not only cover the pure functionality, it covers features like test for isolated and embedded VCs.

Besides the most common distinction between soft cores and hard cores, it makes sense to distinguish between six types that are explained in Table 1.

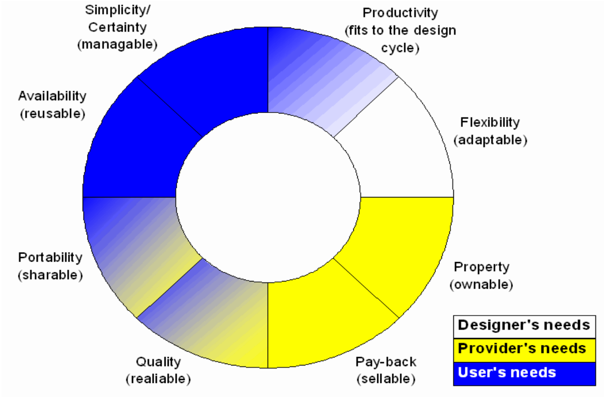
Table 1: Types of IP

Type of IP	**Feature Description**
Hard Cores	Physical layout for a specific process
Soft Cores	Synthesizable netlist that can be targeted to a specific technology
Firm IP	Netlist including parts of the physical placement
Physical Libraries	Building blocks including memory, standard cells and data-paths
Board Libraries	Comprised of LSI, MSI and gates
Software Libraries	Embedded SW functions targeted to a processor

3 IP Sources and Viewpoints

Depending on the point of view, different requirements are in the focus for providing a SoC design. From the viewpoint of the provider, the development of VCs is the most important aspect. But in close relationship to the development four additional items are important: Property, pay-back, quality and portability.

Figure 1: IP Viewpoints



Property and pay-back mainly refer to financial and legal aspects of bringing the IP into the market. The remaining items have to be shared with the users of IP because the level of quality has direct impact on the design, and portability might be important to share the VC among different designs. Availability and simplicity of the component exclusively belongs the IP users. The third point of view is the designers' point of view. He has to enhance his productivity by selecting and using the VC. Flexibility of the VC can enhance its adoptability to similar specifications. Some reuse management systems actively support the retrieval of similar components, if a given specification cannot be fulfilled by the available components. The relationships are visualized in Figure 1.

4 Standardization Initiatives

The domain of design reuse is related to purely technical areas, like the design of a common interface to plug VCs together and it is related to organizational areas like the creation of new business models that support the exchange of VCs.

4.1 Methodology Development

Design reuse can be divided into several areas which cover the technical aspects of VCs. The Virtual Socket Alliance (VSI) tries to cover all aspects by the introduction of seven Development Working Groups (DWG):

- Implementation/Verification
- IP Protection
- Manufacturing Related Test
- Mixed Signal
- On-Chip Buses
- System-Level Design
- VC Transfer

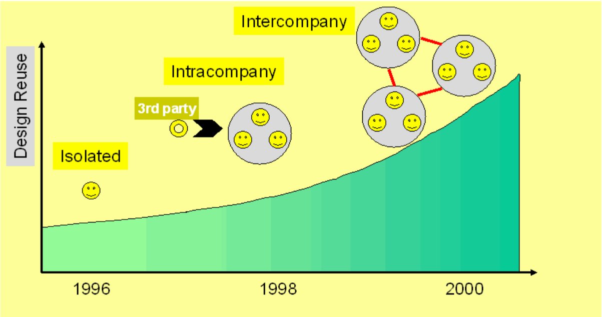
Each group consists of a chair and several members that are interested in that particular topic. All DWGs belong to one technical committee that coordinates the work and strategy of the DWGs. It is important that VSI is not developing a single EDA tool but a set of new or adapted (de facto) standards that will be publicly available after it has passed the internal review process.

4.2 Roadmap and Business Model

The development of a business model for reuse application is very important, since it supports the methodological shift that is caused by the application of reuse techniques and the necessities to apply design for reuse. Initially, this leads from isolated reuse solutions (e.g., [Buet95]) to techniques of centralized or distributed databases and to the embedding of selective reusable system building blocks.

At this stage, a clear (standardized) interface description is required. Third party IP will be integrated from selected partners. Until that point in time, in-house reuse is mainly performed. The final step into an intercompany reuse can be realized if fundamental interfaces are standardized and accepted, integration platform are available and IP business models are installed. Figure 2 shows a roadmap that might be realized during the next years.

Figure 2: Methodological Shift



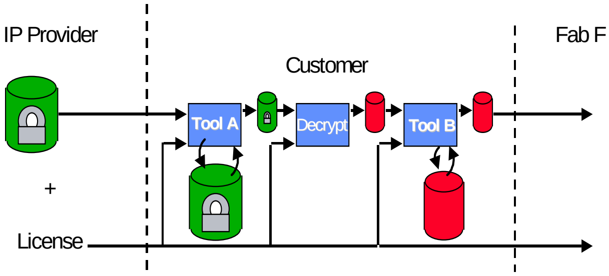
5 IP Protection

The IP protection flow is the first example to demonstrate the importance of methodologies to support the idea of introducing IP in a virtual market. Since IP cannot be protected during the entire design flow, alternative techniques have to be proposed. One promising approach is to combine encrypted IP data with a license that works in parallel to the common design flow.

The IP Protection DWG is chartered to define, document, and demonstrate open, interoperable, standards-based solutions for IP protection which balance the necessary level of security with customer usability of VCs to foster the proliferation of design reuse.

At the beginning of the design phase, the IP provider starts to encrypt his data, because he wants to protect the data also during internal development (cf Figure 3). After the IP has been released for dissemination on the market, the provider transfers the design data and a license to an authorized customer. The customer installs both the data and the license on his site and uses customized EDA tools. Those tools only decrypt the data during internal processing while all intermediate formats are still encrypted.

Figure 3: IP Protection Flow



The existence of tools of type *Tool A* is a prerequisite to maintain data protection. Decode is required, if a tools does not support this kind of licensing (e.g., *Tool B*). In this case, no data protection can be maintained. The basic idea is to accompany the design flow by a IP provider's license until the data can be transferred into the fab. Of course, in this last step the data will not be decrypted.

Three major aspects can be maintained: It is possible to custom the design of the core and its target ASIC by the semiconductor service group, while the core vendor maintains control of its own IP. The vendor integrates its logic into the customer's system logic and he still maintains the control of its IP. Finally, the designer applies standard but customized EDA tools to integrate the vendors core, while the simulation can be performed without restriction.

6 Reuse Management System

As a second example to illustrate the shift to a new paradigm, a comprehensive reuse management system (RMS) will be described. RMS[1] has been developed at FZI Karlsruhe [Seep98a].

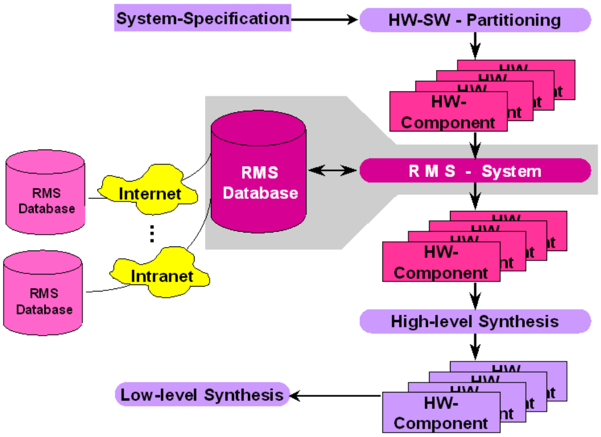
6.1 IP Reuse Flow

RMS is part of a hardware design flow. The base structure of this flow is given in Figure 4. The flow starts at the system specification level. It is possible to split the specification into a hardware and a software part. Since the current model focuses on hardware design, only the hardware-related part is further refined. The partitioning activity offers several hardware components, which are represented by *HW-*

[1] This research is part of EURIPIDES (01M 3036 A) that is supported by the German Ministry (BMBF). EURIPIDES has a MEDEA label (A-407).

Component. These components are processed by RMS and passed to a high-level synthesis system. This system computes a description that can be passed into a low-level synthesis system.

Figure 4: RMS Design Flow

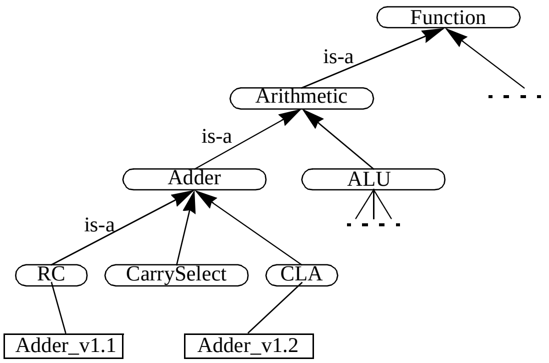


The reuse specific parts of this design flow are *RMS System* and *RMS Database* which might be distributed. A specified hardware component can be replaced by a reusable component of the *RMS Database*, if the specified component and the reusable component is equivalent. Furthermore, the required component can consist of a set of sub-components. According to this structure, the specified component can be divided into a disjunct set of sub-modules. In spite of trying to substitute the complete module by an existing reusable component, it is possible to replace only parts of its structure. RMS proposes several candidates for reuse, which are fitting according to the specified boundary conditions or a set of similar components is proposed that might be automatically adopted. Based on the taxonomy, similarities can be computed with the support of two metrics: asymmetric similarity and conceptual similarity.

6.2 RMS-Taxonomy

The RMS-Taxonomy is strictly hierarchical organized, and therefore, it can be represented by a tree [Seep99]. At the top level, the virtual components (VC) are subdivided into disjunct classes of components, e.g., processors, controllers or buses. Each of this class can be refined into several subclasses. The relationship between an upper and a lower class is provided by a *is-a* semantic [Rumb94]. A VC is classified within the RMS-Taxonomy by assigning it to a leaf node which best characterizes the VC's functionality. In Figure 5, the classified VCs are represented by rectangles, that are contained in a taxonomy tree.

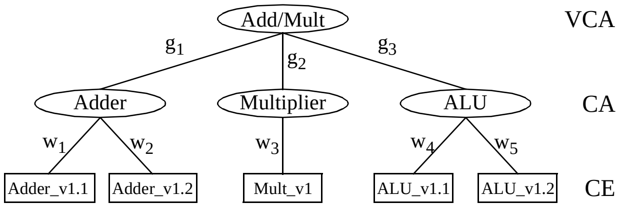
Figure 5: RMS-Taxonomy



6.3 RMS-Classification

The similarity metric used in the extended RMS-Classification is facet-based. Each VC is regarded as a Component Environment (CE) that is associated to a Characteristic Attribute (CA). Weights express the level of similarity between the CEs according to a CA. The granularity of the taxonomy is controlled by Vectors of Characteristic Attributes (VCA) (cf Figure 6). As a part of the architecture, RMS has a very flexible object-oriented data model for managing IPs. As an extension, RMS-Classification offers two new concepts: asymmetric similarity and conceptual similarity.

Figure 6: RMS-Classification



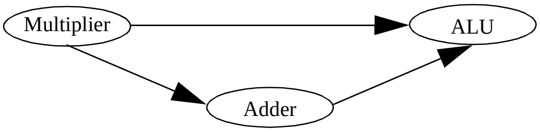
Asymmetric Similarity

For the basic RMS similarity metric, the similarity between reusable components is defined as a symmetric relation. But several asymmetric relationships can occur. For example, an ALU can be used instead of a multiplier but not vice versa. Exactly the same is true for an adder and a ALU or an adder and a multiplier, because a multiplication might be implemented by repeated additions.

If CAs of one VCA (e.g., `Add/Mult`) are considered, which are asymmetric similar to each other, the resulting similarity relationship corresponds to a partial order. This order is equivalent to a Directed Acyclic Graph (DAG). Therefore, each VCA contains a customized DAG that describes the asymmetric similarities between CAs. Those CAs, which are not included in this DAG, are symmetric similar to each CA within this VCA.

Figure 7 presents the DAG of the VCA (`Add/Mult`) for the relationship between `Multiplier`, `Adder` and `ALU`.

Figure 7: Asymmetric Similarities for VCAs



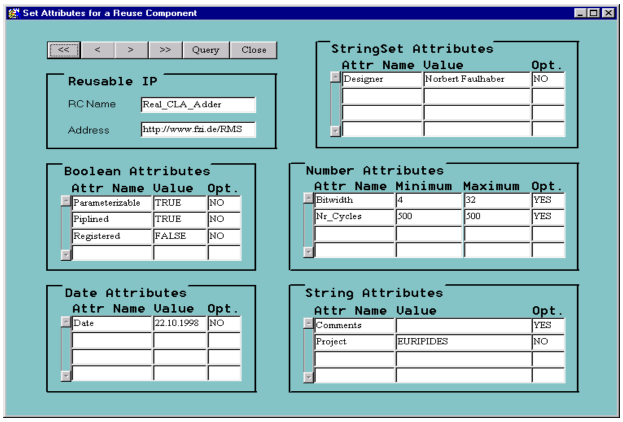
Conceptual Similarity

The conceptual similarity refines the basic RMS similarity metric that was defined between Component Environments (CE), i.e., this is the conceptual distance metric. Conceptual similarity defines the similarity between two CAs related to the same VCA like for `Adder` and `Multiplier`, i.e., the sum of g_1 and g_2 (cf Figure 6). Thus the similarity between CAs is implicated by the similarities of CEs, while maintaining the existing relationships. CAs can be viewed as classes of components. This concept allows the definition of similarities between classes of VCs. It is applied to define similarities between internal nodes of the RMS-Taxonomy.

6.4 VC Retrieval

For retrieval in RMS, the function or the characteristic of a VC has to be specified as a combination of browsing within the RMS-Taxonomy and providing additional input via attribute specification tables. These attributes are specific to a selected taxonomy node. After the search procedure has been initiated, the designer has to select a component from the list proposed by RMS, and in a final step, he can check all attributes specified. It is possible to inspect similar components that are contained in the same list. For each component computed by the retrieval algorithm, a separate attribute window is available (cf Figure 8). With the help of RMS, the designer can easily select appropriate VCs from the RMS reuse database.

Figure 8: Component Retrieval



Reusable IP

RC Name: Real_CLA_Adder

Address: http://www.fzi.de/RMS

StringSet Attributes

Attr Name	Value	Opt.
Designer	Norbert Faulhaber	NO

Boolean Attributes

Attr Name	Value	Opt.
Parameterizable	TRUE	NO
Piplined	TRUE	NO
Registered	FALSE	NO

Number Attributes

Attr Name	Minimum	Maximum	Opt.
Bitwidth	4	32	YES
Nr_Cycles	500	500	YES

Date Attributes

Attr Name	Value	Opt.
Date	22.10.1998	NO

String Attributes

Attr Name	Value	Opt.
Comments		YES
Project	EURIPIDES	NO

7 Conclusion

The development of reuse of IP and virtual components is driven by the investigation of innovative methodologies that fulfil both the application of (de facto) standards and the introduction of tools that accommodate facts to theory. To show the recent stage of development, two typical approaches have been presented: The IP protection flow from VSI and the reuse management system RMS from FZI, which can be regarded as the first comprehensive reuse system for hardware development. Both approaches are going to be evaluated by pilot projects or ongoing application projects.

8 References

[Buet95] Gerhard H. Büttner. “Setting up a Retrieval System for Design Reuse -Experiences and Acceptance-.” *European Design Automation Conference*, pages 575–578, 1995.

[Rumb94] James Rumbaugh et al. *Objektorientiertes Modellieren und Entwerfen*. Carl Hanser and Prentice-Hall, 1994. In German.

[Seep98a] Ralf Seepold. *A Hardware Design Methodology with special Emphasis on Reuse and Synthesis*. Berichte aus der Informatik. Shaker Verlag GmbH Aachen, Ph.D. Thesis, University of Tübingen, 1998. ISBN 3-8265-3417-4.

[Seep98b] Ralf Seepold. “Survey of Reuse.” *MEDEA/ESPRIT Conference HW/SW Codesign*, pages VI–1, September 1998.

[Seep99] Ralf Seepold and Arno Kunzmann (Ed.). *Reuse Techniques for VLSI Design*. Kluwer Academic Publishers, 1999.

[Semi97] Semiconductor Industry Association. “The National Technology Roadmap for Semiconductors.” *Semiconductor Industry Association*, 1997.

[Virt97] Virtual Socket Alliance. “Architecture Document.” Technical Report Version 1.0, Virtual Socket Alliance, http://www.vsi.org, 1997.