Power Contracts for Consistent Power-aware Design

Gregor Nitsche OFFIS - Institute for Information Technology, Germany

***Abstract*—Since energy consumption is one of the most limiting factors for embedded and integrated systems, today's microelectronic design demands urgently for power-aware methodologies for early specification, design-space exploration and verification of the designs' power properties. Most often, these power properties are described by additional power models to extend the functional behavior by the power consumption and power management behavior. Since for consistency high-level estimation and modeling of power properties must be based on a more detailed low level power characterization, we currently develop a contract- and component-based design concept for power properties, called *Power Contract*s, to provide a formal link between the bottom-up power characterization of low-level system components and the top-down specification of the systems' high-level power intent. Formalizing the validity constraints of power models within their *assumption*s, power contracts provide the corresponding formalized power behavior within their *guarantee*s. Hence, being annotated with power contracts, the sub-components of a system can hierarchically be composed according to the concepts of *Virtual Integration* (*VI*), to derive the final system level power contract. Enabling the formal verification of the power models' VI the power contracts allow for a sound and traceable bottom-up integration and verification of power properties.**

I. Introduction

Energy consumption has become one of the most limiting factors for today's embedded and integrated systems. As a consequence, the microelectronic design demands urgently for consistent methodologies which allow for an early specification, design-space exploration and verification of the systems' power properties. To this end, different approaches are developed for high-level power estimation or an automatic synthesis, characterization, abstraction and back-annotation of lower-level power characteristics. Nevertheless - being strongly dependent on future design decisions and low-level parameters, and since system and component power models can only provide a constrained validity, the reliability of such power estimations is strictly uncertain, depending heavily on the correct re-use of power models within a proper embedding environment.

To address this problem of *power closure*, we propose a contract- and component-based design concept, called *Power Contract*s [1], to provide a formal link between the top-down specification of the systems' high-level power intent and the bottom-up power characterization of low-level system components. For that purpose, we apply the ideas of *Heterogeneous Rich Component*s (*HRC*s) [2]–[5] and *Contract Based Design* (*CBD*) [4]–[7] to enable a component-based re-use of reliable power properties in a hierarchical design. While generally aiming towards a comprehensive consistency of power properties across several levels of abstraction our most recent work was focussed on the specification, implementation and verification of *leaf-node* power contracts, meaning those contracts at the lowest abstraction level of the *Virtual Integration*.

In the following, to give an overview of our extra-functional design concept with power contracts, we first outline the underlying basic concepts followed by a draft of our current understanding of power contracts and a rough sketch of the complete methodology in Section II. As a first proof of concept we applied our methodology to the artificial example of an *Advanced Encryption Standard* (*AES*) system, which will be published in the near future. Finally, in Section III we conclude our idea, giving an outlook to our next steps of work.

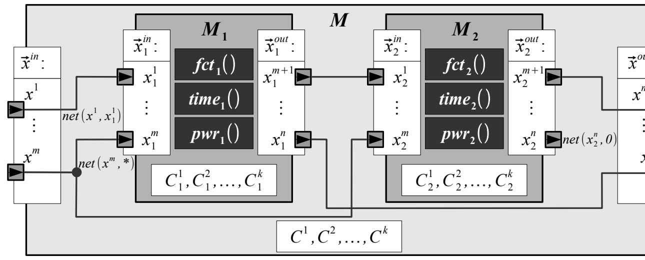

Figure 1. Contract Based Design (CBD): identifiers and basic concept.

II. Basic Concept

HRCs denote structural design elements – i.e. *component*s – which are semantically enriched with *contract*s, being a formal specification over the component's interfaces, which declares *assumption*s on the components' *environment* and *guarantee*s on their externally observable behavior. Hence, the external interaction of an HRC is solely restricted to its explicitly declared interface. Their notion of heterogeneity results from combining the behavioral descriptions of different, functional and extra-functional aspects within the same HRC.

The general idea of HRCs and CBD is summarized in Fig. 1, denoting contracts by C and HRCs by M, additionally indexed by $i \in \mathbb{N}^+$ to refer to the ith HRC of a decomposition of M into n HRCs $\{M_1, \dots, M_n\}$, also called *part*s. For the interface of an HRC M we define χ_M as the set of its directed in- and output variables $x \in \chi_M := \{\vec{x}^{in}, \vec{x}^{out}\}$, called *port*s. To choose only the functional-, timing- or power-specific subsets of M, C, χ and $\vec{x}$, we provide the subscripts *fct*, *time* and *pwr*, corresponding to the internal aspect-specific separation of concerns of the HRC behavior. Finally, we define the HRCs' interconnection network $Net := \{Net_{asm}, Net_{del}\}$ by the sets of its internal *connector*s, consisting of: the *assembly connector*s $Net_{asm} \subseteq \chi^{out}_{M_i} \times \chi^{in}_{M_j}; i, j \in \{1, \dots, n\}; M_i, M_j \in M$ which internally link ports between different parts of the system; and its *delegation connector*s $Net_{del} \subseteq \{\chi^{out}_{M_i} \times \chi^{out}_{M}\} \cup \{\chi^{in}_{M} \times \chi^{in}_{M_i}\}; i \in \{1, \dots, n\}; M_i \in M$ which link up the port of the HRC's parts with the HRC's external ports. The direction of $net(x_{src}, x_{snk}) \in Net$ is defined by position, naming the source port x_{src} in front of the reader's sink x_{snk}. Finally, $net(x_{src}, *)$ and $net(*, x_{snk})$ denote a multi-point connection from a common source respectively a multi-point connection to a common sink with $net(x_{src}, 0)$ or $net(0, x_{snk})$ denoting open ports.

Based on that, the contracts C of M are formally defined as triples $C := (A, B, G)$ of *strong assumption*s A, delimiting the component's permissible input space by predicates over $\vec{x}^{in}$, the *weak assumption*s B, performing another division to distinguish more constraint subspaces and guarantees G, denoting predicates over $\vec{x}^{out}$, being assured if the environment satisfies the associated assumptions. Hence, C is semantically interpreted as $[\![C]\!] := ([\![A]\!] \wedge [\![B]\!]) \Rightarrow [\![G]\!]$, with A, B and G being time bounded LTL or CTL properties, representing sets of *timed traces* $S_A(\vec{x}^{in}), S_B(\vec{x}^{in})$ and $S_G(\vec{x}^{out})$

over the I/O variables χ_M of M. Declaring the type of a variable x to be $\nu(x) \in \{\mathbb{B}, \mathbb{Z}, \mathbb{N}, \ldots\}$ and declaring the notion of *time* as a discrete but infinitely increasing variable $t \in \mathbb{N}_0^+$, a *timed trace* $s_x(t)$ is a discrete sequence of *event*s $\{e(x, t_0), e(x, t_1), \ldots\} \in S_x := \{x \to [\mathbb{N}_0^+ \to \nu(x)]\}$, mapping the variable x to its *value*s $v(x, t_i) \in \nu(x); i \in \mathbb{N}_0^+$ for each point of time. Considering all contracts C_{asp}^i of all aspects $asp \in \{\text{fct, time, pwr}\}$ a complete property specification C of M is defined as $C := \bigwedge_{asp} \bigwedge_{i=0}^{n_{asp}} C_{asp}^i$

To extract an expression of the effect of the assumptions, guarantees or even complete contracts on only a subset of ports, the *restriction function* $\downarrow_X$ denotes the restriction of these constraints to solely the subset X of their original variables. Additionally, ρ_i and ρ denote the *port substitution* functions, of which ρ_i identifies the port variables $x_i^{in}, x_i^{out} \in \chi_{M_i}$ of a part M_i with the corresponding assembly and delegation connectors $net(x_i^{in}, *), net(*, x_i^{out}) \in Net$; and ρ identifies the external ports $x^{in}, x^{out} \in \chi_M$ of the system M with the corresponding delegation connectors $net(x^{in}, *), net(*, x^{out}) \in Net$.

Hence, contracts allow to explicitly relate a formalized bottom-up characterization of a component's power behavior with the power model's validity constraints, a composition and abstraction would assume to be satisfied without verification. Hence, CBD enables to formally check for:

- *compatibility*: $G_{M_{src}} \downarrow_{x_{src}} \rho_{M_{src}} \Rightarrow A_{M_{snk}} \downarrow_{x_{snk}} \rho_{M_{snk}}$ between the connected components of a system;
- *refinement*: $C' \Rightarrow C$ of a system M's specification C w. r. t. its component-based bottom-up composition by n parts $\{M_1, \ldots, M_n\}$, specified by their contracts C_i and logically composed to the *VI* $C' := ((\bigwedge_{i=1}^{n} C_i \rho_i)\, \rho \downarrow_{\chi_M})$.

Applying the concepts of HRCs and CBD to bottom-up leaf-node power models our goal is to formally ensure the correct re-use of power models within a consistent, power-aware design flow, improving power closure. Our basic idea for that design and verification flow is outlined in Fig. 2, covering:

1) the structural decomposition of the initial HRC with possibly a refined partitioning of its initial contracts;
2) the implementation of the HRC's parts;
3) the formal bottom-up characterization of the parts' functional and extra-functional behavior in terms of contracts;
4) the *satisfaction* checking between the parts' contract based bottom-up characterization and their specification;
5) the compatibility checking between the connected ports;
6) the virtual integration to a composed top-level contract;
7) the refinement checking between the composed top level contract and those of the initial specification.

Considering the steps $2-4$ we currently present an approach using *UPPAAL* to specify top-down leaf-node power contracts which are verified against systematically obtained, bottom-up leaf-node power models. For that, our current notion of power contracts follows Fig. 3, addressing the most relevant factors of dynamic power consumption. As a result, we provide verified bottom-up power contracts for compatibility and refinement checking within the hierarchical VI process, enabling a sound and traceable methodology for re-using power models.

III. Conclusion

Due to the need for power closure we investigate heterogeneous rich component- and contract based design to build a consistent flow for power-aware system design. Integrating the functional, timing and power aspects of a bottom-up component characterization, we derive formal HRCs in UPPAAL, enabling

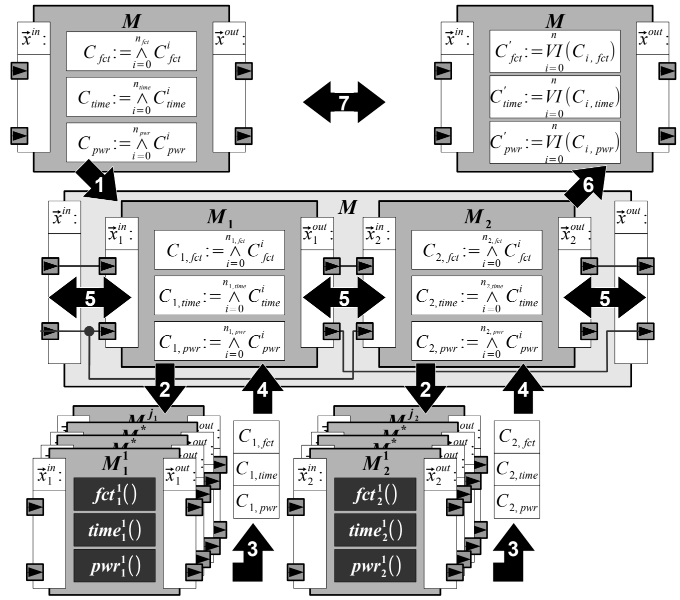

Figure 2. Basic idea of the design steps within a power-aware design flow with power contracts.

Power Contract		
HRC Reference		
A:	Implementation	Technology, Architecture, Power Domain
B:	Functional Mode	I/O Values
	Power Mode	Voltage, Frequency
G:	Power Consumption	State Power, Power Gradient, Minimum Power, Maximum Power, Average Power

Figure 3. Content and structure of a power contract.

for an exhaustive verification between specification and implementation, based on contracts. Applying power contracts to the example of a composed AES system, we successfully analyzed an initial proof of concept, allowing for the leaf-node verification of power contracts in UPPAAL.

References

[1] G. Nitsche, K. Grüttner, and W. Nebel, "Power contracts: A formal way towards power–closure?!" in *Proceedings of the 23rd International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS'13)*. Karlsruhe: IEEE, 2013.

[2] B. Josko, Q. Ma, and A. Metzner, "Designing embedded systems using heterogeneous rich components," in *Proceedings of the INCOSE International Symposium 2008*, 2008.

[3] W. Damm, H. Dierks, J. Oehlerking, and A. Pnueli, "Towards component based design of hybrid systems: Safety and stability," in *Time for Verification Essays in Memory of Amir Pnueli*, ser. Lecture Notes in Computer Science, Z. Manna and D. A. Peled, Eds. Springer-Verlag, Berlin, Heidelberg, 2010, vol. 6200.

[4] W. Damm, H. Hungar, B. Josko, T. Peikenkamp, and I. Stierand, "Using contract–based component specifications for virtual integration testing and architecture design," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*, M. Lukasiewycz, S. Chakraborty, and P. Milbredt, Eds., Grenoble, France, 2011.

[5] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming dr. frankenstein: Contract-based design for cyber-physical systems," *European Journal of Control*, vol. 18, no. 3, 2012.

[6] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger, and K. Larsen, "Contracts for systems design," Research Centre Rennes – Bretagne Atlantique, Rennes Cedex, Tech. Report RR-8147, 2012.

[7] H. Hungar, "Components and contracts: A semantical foundation for compositional refinement," in *Tagungsband MBEES: Modellbasierte Entwicklung eingebetteter Systeme 2012*, 2012.