

**Kurzfassung** – Die Komplexität von Hardware-/Software-Systemen (HW/SW-Systemen) erhöht sich, getrieben von den Fortschritten der Technologie, immer weiter. Die schnelle technologische Entwicklung führt zu kurzen Produktlaufzeiten. Dies wiederum erfordert möglichst kurze Entwicklungszeiten von Produkten, um so die Lebenszeit der Produkte zu verlängern. Es ist deshalb zwingend erforderlich, immer neue Methoden zu entwickeln, um einen schnellen und fehlerfreien Entwurf von Systemen zu ermöglichen. Die vorliegende Arbeit beschäftigt sich in diesem Kontext mit verschiedenen Aspekten des Entwurfs von HW/SW-Systemen. Der thematische Schwerpunkt liegt dabei auf Methoden zur Spezifikation und zur Validierung von HW/SW-Systemen.

Zunächst werden Kriterien und Anforderungen zur Spezifikation von HW/SW-Systemen erarbeitet, die schließlich in der Entwicklung der Systemsprache *CoLa* münden. Bestehende Sprachen, wie *C* oder *VHDL*, eignen sich entweder besser für die Spezifikation von Software oder aber von Hardware. Dies führt dazu, daß bestimmte Sprachkonstrukte nur in einem Systemteil implementiert werden können. Die Partitionierung wird so durch die Spezifikation bereits teilweise vorweg genommen. *CoLa* hingegen ist für Hardware und Software gleichermaßen gut geeignet und unterstützt dadurch einen unvoreingenommenen Entwurfsprozeß von HW/SW-Systemen. Ein Vorteil ist insbesondere die einheitliche Formulierung von Nebelläufigkeit, die sowohl in parallele als auch in serielle Komponenten umgesetzt werden kann. Darüber hinaus ermöglichen Konstrukte zur Spezifikation von Zeitbedingungen die Formulierung von Echtzeitanforderungen. Aufgrund der leistungsfähigen Datenstrukturen, der Möglichkeit zur Verwendung von beliebigen Rückgabewerten und der Nutzung von Rekursion, profitiert *CoLa* von den Vorteilen funktionaler Sprachen. Die für HW/SW-Systeme wichtige Beschreibung von Systemzuständen wird mit Hilfe von speziellen statischen Variablen und der Möglichkeit zur direkten Darstellung von endlichen Automaten erreicht. Darüber hinaus gibt es in *CoLa*, anders als zum Beispiel in *VHDL*, nur ein universelles Konstrukt zur Darstellung von Hierarchien. Dieses kann zur Definition von Funktionen, zur Bildung von strukturellen Einheiten und zum Einfügen vordefinierter Komponenten verwendet werden.

Um *CoLa* in der Praxis zu erproben, wird mit der *CoLaToolbox* der Prototyp eines Entwurfssystems für HW/SW-Systeme vorgestellt. Das Rückgrat der *CoLaToolbox* bildet eine leistungsfähige interne Repräsentation, die aus einer Beschreibung in *CoLa* gewonnen wird und jederzeit wieder in *CoLa* zurück gewandelt werden kann. Auf diese Weise wird eine umfassende Modellkontinuität erreicht, an der es vergleichbaren Entwurfswerkzeugen mangelt.

Ein besonderes Augenmerk wird auf die Simulation gelegt. Zu diesem Zweck wird eine Anbindung der *CoLaToolbox* an die Simulationsumgebung *HADES* verwirklicht. Sie ermöglicht die automatische Generierung eines Simulationsmodells zur interaktiven Simulation des Systems und seiner Umgebung. In die Simulation können bereits implementierte Systemteile integriert werden, um auf diese Weise einzelne Systemteile separat zu testen.

Schließlich wird mit *DINAKO* ein exemplarisches Zielsystem präsentiert, innerhalb dessen ein Mikroprozessor über ein Bussystem mit einer rekonfigurierbaren Hardware verbunden wird. Es erlaubt aufgrund des variablen Hardwareteils die Untersuchung von verschiedenen Entwurfsalternativen in der Praxis. Mit *DINAKO* ist es möglich, gewählte Partitionierungen zu testen, die Performanz unterschiedlicher Partitionierungen miteinander zu vergleichen und die Zuverlässigkeit von Schätzungen zu überprüfen. Dabei ist es insbesondere möglich, den tatsächlichen Kommunikationsaufwand zwischen den Systemteilen unter realen Bedingungen zu ermitteln.

**Summary** – Driven by the progress of technology, the complexity of hardware/software systems increases more and more. This progress also leads to short product cycles and requires very short product development times in order not to miss the market window. Therefore, it is mandatory to develop ever new design methods in order to enable a fast and error free system design. Previous work in this context has concentrated on many different aspects of the design process. The emphasis of this work is on methods for specification and validation of HW/SW systems.

First, requirements for the specification of HW/SW systems are identified and collected. These criteria are then used in the development of a new system language, *CoLa* (Codesign Language). Most existing languages, like *C* or *VHDL*, are usually better suited either for the specification of software or hardware. This implies that certain language constructs can be implemented only in one system part (hardware or software) and the partitioning is already anticipated by the specification. *CoLa*, however, is suitable for hardware and software equally well and supports an impartial design process of HW/SW systems. One particular advantage is the uniform formulation of concurrency, which can be translated into both parallel or serial components. Additionally statements for timing constraints allow the specification of real time requirements. Also, *CoLa* profits from the advantages of functional languages. It possesses efficient data structures, recursion can be used, and functions are allowed to return multiple return values. The modelling of states, important for HW/SW systems, is achieved by static variables or a direct representation by finite state machines. *CoLa* uses a single but universal concept for the representation of hierarchy. This can be used for the definition of functions, for the description of structures and for inserting pre-defined library components.

In order to test *CoLa* in practice, the *CoLaToolbox* is introduced as a prototype of a design system for HW/SW systems. An efficient internal representation, compiled from a description in *CoLa*, forms the backbone of the *CoLaToolbox*. The representation can always be translated back to a *CoLa* description. In this way a global model continuity is achieved which other design tools usually lack.

Special attention is placed on simulation. For this purpose the *CoLaToolbox* includes an interface for the simulation environment HADES. It enables the automatic generation of a simulation model for the interactive simulation of the system and its environment. Into the simulation system, those parts already implemented can be integrated, in order to test such system parts separately in this way.

Finally, an exemplary target system DINAKO is presented which consists of a microprocessor connected via a PCI bus system to a FPGA-based reconfigurable hardware. Due to the variable hardware section, it permits rapid exploration of different design alternatives. With DINAKO, it is possible to test selected partitionings, to compare their performance and to check the reliability of estimates. In particular, the actual communication overhead between the system parts can be determined under real conditions.