Simulation and Analysis of Embedded DSP Systems using MASIC Methodology

Abhijit K. Deb, Johnny Öberg, Axel Jantsch
Department of Microelectronics and Information Technology
Royal Institute of Technology, 164 40 Kista, Sweden
Email: abhijit | johnny | axel @ imit.kth.se

Abstract

MASIC, short for Maths to ASIC, models embedded DSP systems using grammar based technique. This paper presents a simulation and analysis technique for the MASIC model of a system. We build a Petri Net (PN) representation of the MASIC model to simulate the communication architecture. It helps to evaluate the effects of architectural decisions early in the design cycle. The correctness of the protocol description is analyzed using PN based boundedness and conservation analysis.

1. Introduction

Modeling in MASIC begins at the functional level where individual DSP functions are developed in a C or MATLAB like environment. At this stage, design decisions are primarily algorithmic. The output of this phase is a set of DSP functions in C. The C functions do not have any notion of time and nor the way to exchange data among them. MASIC uses grammar specification to create communication channels among the functions and other architectural decisions [2]. The communication (expressed in grammar) and computation (expressed in C) are cleanly separated in the model. An embedded core is viewed at two different layers. The primary intention of the model is to focus on the interface between the host and the core [3]. The inner layer contains the DSP function in C, developed in the functional modeling phase. The outer layer provides the Bus Functional Model (BFM) of the core. The BFM is an abstract model of the core interface that generates the interface transactions. The core interface provides the means to transfer data. Once the data is transferred, C-functions are used to perform the desired computation inside the block.

As the model is refined, the abstract channels of communication turn out to be wires and buses, necessarily arbitrated by a global controller. The bus protocol and the SoC communication architecture influence the performance of an embedded system due to the synchronization overhead [4]. Effects on performance of HW/SW systems, due to shared memory access and DMA block size where system operation is dominated by memory accesses, have been reported in [5]. An accurate analysis and simulation of the model is possible if it is compiled to a RT level using the MASIC compiler. However, RT level is too detailed to be efficient enough during exploration of the design space.

Analytical techniques have been proposed to evaluate the system performance due to the communication architecture [4][6]. Passerone et al. have presented a co-simulation technique for trade-off analysis of embedded systems [7]. A design space exploration method by communication analysis based on Monotone abstraction has been reported in [8].

In this paper we present a technique to simulate the communication architecture of the MASIC model at the clock true level.

2. Petri Net Based Analysis

In MASIC, an embedded system is viewed as a set of communicating FSMs. The problem of analyzing the set of communicating FSMs is that the combination of state machines is complex, requiring a composite state with components of both sub-machines, a cross-product machine. A memory would require as many states as the number of values that can be stored at each location raised to the power of number of locations [9]. The number of states often has a good correlation to the complexity. However, composition is simple in a PN [1]: a serial composition is achieved simply by overlapping the output places of the first net with the input places of the second net. For a parallel composition, it involves duplicating the input tokens, which represent the input symbols and feeding those to each component PN.

We convert the set of communicating FSM into the PN model. This PN model is simulated at the clock true level that reveals the transactions of the communication architecture. Hence we get the effects of an architectural decision early in the design cycle, that is before the design is compiled to RT level. The design flow in MASIC is shown in Figure 1.

1530-1591/03 $17.00 © 2003 IEEE

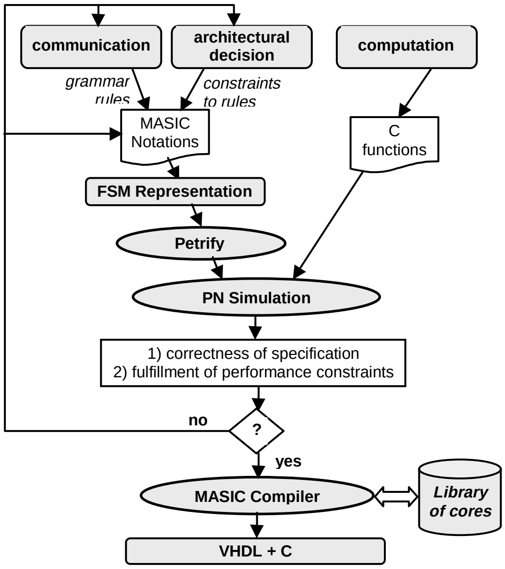

Figure 1: MASIC design flow

2.1 Building the PN Model

Each state of the communicating FSMs is represented as a place in the PN. We call them *state-places*. A token in one of these state-places symbolizes the present state of the machine. Signals used for handshaking between FSMs are called *synchronization signals*, and signals used for storage (memory element) and interconnect (buses) are called *data signals*. Each synchronization signal is represented by two places: a *high-place* and a *low-place*. A token in the high or the low-place of a signal respectively represents the high or low state of that signal. Data signals are viewed as *placeholders* for data values. They are not a part of the net instead they are updated with new value when an assignment is made.

To have an abstract representation of time we introduced *clock place*, where tokens appear with respect to time and disappear with the subsequent firing. The appearance of a token in a clock-place symbolizes the appearance of a clock edge and advances the simulation time. Transitions are used to model change of the state of an FSM or a signal. To model a combinational change, a transition without the clock-place as one of its input places is used. Sequential changes are modeled by a transition, which has a clock-place as one of its input places. Arcs are drawn between input places to transitions and transitions to output places, which denote the causal relation described in the MASIC description.

2.2 Simulating and Analyzing the Net

Clock place is used to sequence events with time and advance the simulation time. Tokens are placed in the clock place so that all enabled transitions at a given time can fire. Non-determinism occurs if the firing of one transition disables another enabled transition. FSMs are supposed to work deterministically. If non-determinism happens that reveals an erroneously specified protocol.

The PN model is also used to analyze the correctness of specification. A token must exist in one of the places (high-place or low-place) of a synchronization signal. Both of the places can not be empty and there can not be more that one token in a place at any instance of time. If they do, that clearly reflects an erroneously specified protocol. Thus checking the conservation and boundedness properties of the PN we can verify one aspect of the correctness of a communication protocol.

3. Conclusion

The PN simulation shows clock true transactions of the communication architecture and helps to make design decision early in the design cycle. Besides, it provides the PN based techniques to analyze the grammar based specification.

4. References

[1] J. Peterson, *Petri Net Theory and the Modeling of Systems*. Prentice Hall, Englewood Cliffs, NJ, 1981.

[2] A.Hemani, A.K.Deb, J.Öberg, A.Postula, D.Lindqvist and B. Fjellborg, "System Level Virtual Prototyping of DSP SoCs Using Grammar Based Approach". *Design Automation for Embedded Systems*, vol.5, no.3, pp.295-311, Aug 2000.

[3] A. K. Deb, A. Hemani, J. Öberg, A. Postula, and D. Lindqvist, "Hardware Software Codesign of DSP Systems using Grammar Based Approach". *In Proc. of 14th Int. Conf. on VLSI Design*, pp.42-47, India, January 2001.

[4] S. Dey and S. Bommu, "Performance analysis of a system of communicating processes". *In Proc. of ICCAD*, pp. 590-597, 1997.

[5] M. Lajolo, A. Raghunathan, S. Dey, L. Lavagno, A. Sangiovanni-Vincentelli, "A Case Study on Modeling Shared Memory Access Effects during Performance Analysis of HW/SW Systems". *Int. Workshop on Hardware-Software Codesign*, pp. 117-121, Seattle, March 1998.

[6] K. Lahiri, A. Raghunathan and S. Dey, "System-Level Performance Analysis for Designing On-Chip Communication Architectures". *IEEE Trans. On Computer Aided Design*, vol. 20, no. 6, pp768-783, June 2001.

[7] C. Passerone, L. Lavagno, C. Sansoe, M. Chiodo, A. Sangiovanni-Vincentelli, "Trade-off evaluation in embedded system design via co-simulation", *In Proc. of ASP-DAC*, pp. 291-297, 1997.

[8] H. Hsieh, F. Balarin, L. Lavagno, A. Sangiovanni-Vincentelli, "Efficient methods for embedded system design space exploration", *In Proc. of DAC*, pp. 607 -612, 2000

[9] S. Edwards, L. Lavagno, E. A. Lee and A. Sangiovanni-Vincentelli, "Design of Embedded Systems: Formal Models, Validation and Synthesis" *Proc. of IEEE*, vol. 85, no. 3, pp. 366-390, Mar 1997