Quantitative Analysis of Transaction Level Models for the AMBA Bus

Gunar Schirner, Rainer Dömer
Center of Embedded Computer Systems
University of California, Irvine
hschirne@uci.edu, doemer@uci.edu

Abstract

The increasing complexity of embedded systems pushes system designers to higher levels of abstraction. Transaction Level Modeling (TLM) has been proposed to model communication in systems in an abstract manner. Although being widely accepted, TLMs have not been analyzed for their loss in accuracy.

This paper will analyze and quantify the speed-accuracy tradeoff of TLM using a case study on AMBA, an industry bus standard. It shows the results of modeling the Advanced High-performance Bus (AHB) of AMBA using a set of models at different abstraction levels. The analysis of the simulation speed shows improvements of two orders of magnitude for each TLM abstraction, while the timing in the model remains accurate for many applications.

As a result, the paper will classify the different models towards their applicability in typical modeling situations, allowing the system designer to achieve fast and accurate simulation of communication.

1. Introduction

System-On-Chip (SoC) design faces a gap between the production capabilities and time to market pressures. The design space, to be explored during SoC design, grows with production improvements, while at the same time shorter product life cycles force an aggressive reduction of the time-to-market. Addressing this gap has been the aim of recent research work. As one approach, abstract models have been introduced to tackle the design complexity.

Fast simulation capabilities are required for coping with the design space complexity; especially during early stages of the design process. To addresses this need, Transaction Level Modeling (TLM) has been proposed [7]. TLM abstracts the communication in the system to whole transactions, abstracting away low level details about pins, wires and waveforms. This results in models that execute dramatically faster than synthesizable, bit-accurate models.

TLM, however trades off speed with decreased accuracy. While TLM has been generally accepted as one solution to SoC design, this tradeoff however, has not been quantitatively analyzed. This paper will quantify and analyze the performance gains of TLM and its accuracy tradeoff. Our analysis is based on a case study of the Advanced High-performance Bus (AHB) of AMBA [2].

In this paper we will first introduce the main features of the AHB. Then, we will propose a set of models with different levels of abstraction and describe their design. We will measure the implemented models in an experimental setup and analyze their results. After discussing analysis results, we will conclude the paper with a classification of the models suitable for particular application types.

2. Related Work

System level modeling has become an important issue, as a means to improve the SoC design process. Languages for capturing such models have been developed (e.g. SystemC [7], SpecC [5]). Capturing and designing communication systems using TLMs [7] has received attention.

Sgroi et al. [12] address the SoC communication with an Network-on-Chip approach. They propose partitioning the communication into layers following the OSI structure.

Siegmund and Müller [13] describe with SystemC^{SV} an extension to SystemC, and propose SoC modeling three different levels of abstraction: the physical description at RTL level, a more abstract model for individual messages, and a most abstract level that deals with transactions.

Coppola et al. [4] propose an abstract communication modeling, present the IPSIM framework and show its efficient simulation, however no accuracy analysis.

Gerstlauer et al. [6] describe a layered approach and propose models that implement an increasing number of OSI [8] layers. They have shown speedup of at most 100x, however the accuracy analysis is very limited.

In [3] Caldari et al. describe the AMBA 2.0 bus system at two abstraction levels: a bus functional model (BFM) at RT level, and a TLM. Their TLM reached a speedup of 100x over the BFM, an accuracy analysis is not presented.

Pasricha et al. [9] describe an AMBA 2.0 model that is cycle count accurate at transaction boundaries. It reaches twice the speed of the BFM.

3. Introduction to the AMBA Bus

ARM defined with the Advanced Microprocessor Bus Architecture (AMBA) [2], a widely used on-chip bus system standard. It contains a group of busses, which are used hierarchically as shown in Figure 1. This paper focuses on the Advanced High-performance Bus (AHB), a system bus designed for connecting high-speed components including ARM processors.

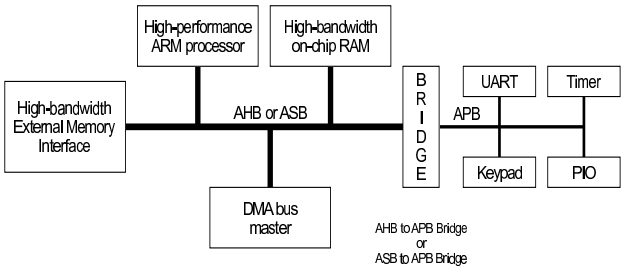


Figure 1. AMBA bus architecture (Source [2]).

The AHB is a multi-master bus that operates on a single clock edge. High performance is achieved by a pipelined operation that overlaps arbitration, address, and data phases, and by the usage of burst transfers. Split and retry transfers allow the slave to free the bus if the requested data is temporary unavailable. The AHB also employs a multiplexed interconnection scheme to avoid tri-state drivers.

4. Modeling

A layered architecture was chosen in order to cope with the communication complexity. Following the ISO OSI reference model [8], the AHB specification falls within the second layer, the data link layer. For modeling of the AHB, the media access control (MAC) and the protocol sublayer are considered, as well as the physical layer.

The OSI layer definition is based on functional concerns. An alternative view, suitable for describing the models, focuses on the granularity in which user data is handled. The *media access layer* provides a transmission service for a contiguous block of bytes, called a *user transaction*. This layer divides the arbitrary sized user transaction into smaller bus transactions observing the bus addressing rules and transfers these byte blocks using the protocol layer. The *protocol layer* transfers data as *bus transactions*, which are bus primitives (e.g. bytes, words, or 4 word burst). It uses the *physical layer* services, which provide a *bus cycle* access to sample and drive individual bus wires.

Figure 2 shows how the above defined data granularity levels can be analyzed with respect to time. A user transaction is successively split into the smaller elements: bus transactions and finally bus cycles.

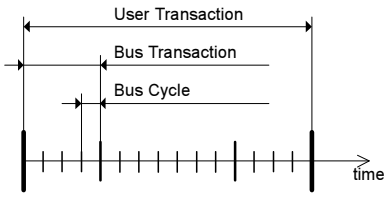


Figure 2. User Transaction Decomposition.

Using a system level modeling approach, we have modeled each layer of the AHB as a separate channel using a system level design language (SLDL). Overall, we have considered 9 different models. For space reasons, we limit the set of models in this paper to five, namely two TLMs, two ATLMs and one BFM. Note that we have introduced an intermediate level, the Arbitrated Transaction Level Model (ATLM), between the known TLM and BFM levels.

4.1 Transaction Level Model (TLM)

The TLM is the most abstract model; it only implements the media access layer. The user data, handled at the user transaction granularity, is transferred regardless of its size in one chunk using a single *memcpy*. Timing is simulated by a single *waitfor* statement, covering the whole user transaction. Arbitration is not modeled. Instead, concurrent access is resolved using a semaphore once per user transaction. Due to using a semaphore, the contention resolution depends on the simulation environment.

Two variances of the TLM where defined for evaluation purposes. The variance TLM (a) implements the above described contention resolution. The TLM (b) does not implement any contention resolution. It unrealistically allows many masters to simultaneously access the same bus.

4.2 Arbitrated Transaction Level Model (ATLM)

The ATLM simulates the bus access with a bus transaction granularity (AHB bus primitives), at the protocol layer level. It uses the MAC layer implementation of the later described BFM to split user transactions into bus transactions.

The ATLM accurately models priority based arbitration for each bus transaction. We implemented the arbitration without an own flow of execution to maximize simulation performance. However, this model is not pin accurate and not cycle accurate in all cases.

We have implemented two variances of the ATLM that differ in the time frame to collect arbitration requests. On an idle bus, the ATLM (a) collects requests for one clock cycle before making a decision. The ATLM (b), on the other hand, makes the decision immediately after receiving the first request. Both variances behave identical when the bus is busy: requests are collected while a bus transaction is active and the highest priority master continues after that.

4.3 Bus Functional Model (BFM)

The BFM is a synthesizable, bus cycle accurate and pin accurate bus model. It implements all layers down to the physical layer and covers all timing and functional properties of the bus definition. It handles arbitration per bus transaction and verifies the bus grant on each cycle of a burst. We implemented additional active components, such as multiplexers, an arbiter and an address generator, for correctly modeling the bus architecture.

Figure 3 summarizes the described models. It shows for each model the lowest layer that it implements and the granularity at which it handles data and arbitration. The user transaction decomposition figure is superimposed to visualize the granularity.

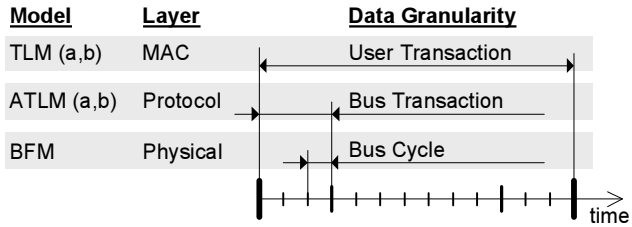


Figure 3. Model Summary.

We have functionally validated all described models. We validated the cycle count timing of the BFM against the standard for the all bus primitives and compared with the example waveforms in [2, 1]. We ensured that all abstract models show the correct timing in a single master setup.

5. Measurements and Quantitative Analysis

In this section, we will analyze the usability of the implemented models by examining two aspects. First, we will measure *simulation performance*, since a performance gain is the main premise of TLM. Second, we will evaluate the *timing accuracy*. Weighting the speed gain against the accuracy loss allows the designer to decide for a speed/accuracy tradeoff suitable for a particular design stage.

5.1 Performance

We examined the simulation performance of each model in a scenario with one master and one slave. User transactions are transferred repeatedly, without any delay in between. We measured the simulation time (also referred to as real time or wall clock time) for all transfers and computed the simulated bandwidth. All tests have been performed on a Pentium 4 at 2.8 GHz.

The measurements (Figure 4) confirm the TLM expectations: the simulation speed increases significantly with abstraction. The performance rises with each TLM abstraction by two orders of magnitude. However, no significant performance difference exists between the variances within the

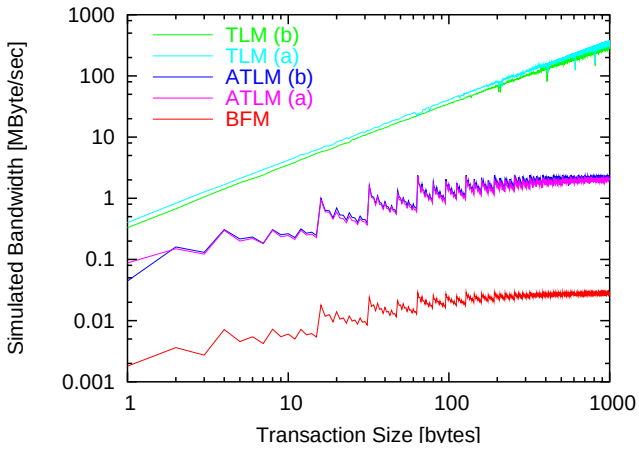


Figure 4. Simulated bandwidth.

two ATLMs and the TLMs. The additional abstractions of the (b) variances do not yield a speed improvement. Table 1 compares the performance for transferring 512 bytes.

Feature	BFM	ATLM	TLM
Simulation Time [ms]	16.75	0.2137	0.00246
Sim. Bandwidth [MByte/s]	0.03	2.29	198
Rel. Speedup over BFM	1	78	6802

Table 1. Performance of a 512 byte transfer.

As expected, the TLMs execute the fastest. Their execution time is independent of the transaction size, since a constant number of operations is executed for each transfer (one *memcpy* and one *waitfor*). Hence, their graphs are linear. The ATLMs are two orders of magnitude slower due to the finer granularity of the modeling individual bus transactions for data and arbitration. Starting with the ATLMs, the graphs exhibit a saw tooth shape due to the non linear split of user transactions into bus transactions (e.g. 3 bytes are transfered in 2 bus transactions: byte + short, whereas 4 bytes are transfered 1 bus transaction: a word). The BFM is again two orders of magnitude slower than the ATLMs, since modeling the individual wires and the active components (e.g. multiplexers) requires an additional effort.

5.2 Accuracy

In the previous section we have quantified the performance gain of our abstract models. Now we will evaluate the accuracy limitations, that the designer has to accept for achieving the higher simulation speed. An accuracy statement depends heavily on the environment, the application at hand and the prediction goal. This section shows the analysis results for a generic test setup with the focus on two prediction goals: (a) the *application latency* due to communication, and (b) the influence of the communication to

the overall *application finish time*. Two different operating modes of the AHB will be analyzed: *locked transfers* and *unlocked transfers*. In the latter one, a burst in transmission may be preempted by a higher priority transfer.

5.2.1 Test Setup

We have used a generic test setup with two masters and two slaves. Each master transfers a predefined set of 5000 user transactions that vary linear randomly in the base address, size and the delay to the next transaction (simulating the application's computation). The start time and the duration is recorded for each individual user transaction and each master. The test is repeated for each implemented bus model. Since each model transfers the same set of user transactions, their results are comparable and can be analyzed.

The models differ in the granularity of data and arbitration handling. We therefore expect a significant change in accuracy with bus contention. Hence, we repeated the described test for different levels of bus contention. However, the contention can not be controlled directly. Instead, we changed the maximum delay between user transactions for each test run. This results in a changed bus utilization per master. Since two masters share the same bus in our setup, the utilization correlates to the amount of bus contention.

We measured the resulting amount of bus contention using the BFM. For each clock cycle, we measured whether one or two user transactions where active (ie. if one or two applications where blocked for completion of a transaction). For this paper, we define the bus contention as the percentage overlap between user transactions, as shown in Figure 5.

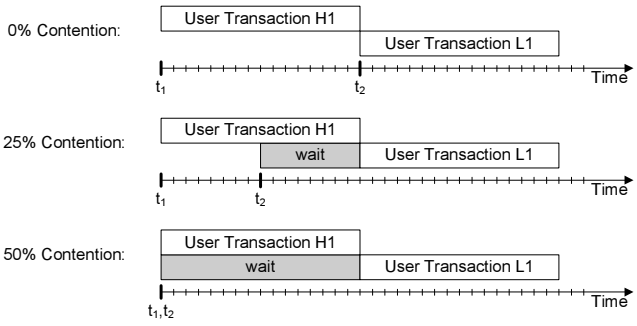


Figure 5. Bus contention.

5.2.2 Analysis for Locked Transfers

We first focus on the operation mode of locked transfers, where a burst can not be preempted by a higher priority master. Independent of the operation mode, the duration of an individual user transaction is an important measure for predicting the application latency due to bus access. Therefore, we evaluate in a first step the accuracy of the models with respect to the transfer duration. For this purpose, we define the error of an individual user transaction as:

$$\begin{aligned} d_{std}: &\quad \text{duration as per AHB standard} \\ d_{test}: &\quad \text{duration in model under test} \\ error_i &= 100 * \frac{|d_{test} - d_{std}|}{d_{std}} \end{aligned} \tag{1}$$

Figure 6 shows the average timing error over a range of bus contention for the high priority master when using locked transfers. The BFM and the ATLM (a) perform accurately over the whole range of bus contention (their graphs lie on top of the x-axis). Since the test setup uses only locked transfers, no arbitration test is needed within a bus transaction. The features abstracted away in the ATLM (a) are not exercised. The ATLM (b), which makes an immediate decision and does not collect further requests, shows an inaccuracy of up to 18%. It may mispredict bus access when the two masters attempt a bus access within the same clock cycle.

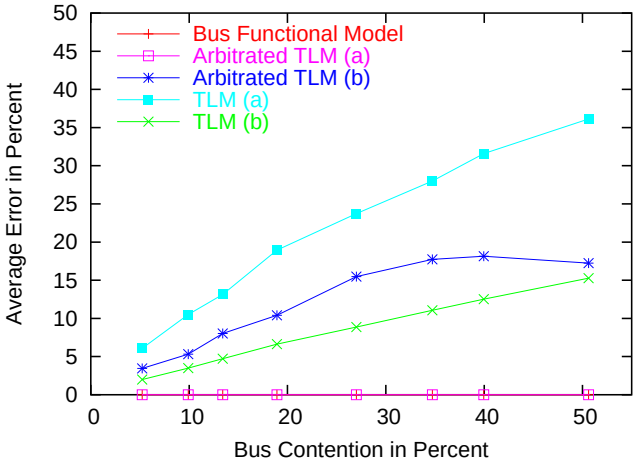


Figure 6. Locked accuracy by duration.

The TLM (a), which handles contention resolution on user transaction level with a semaphore, performs least accurate due to the coarse grained decision that is made independent of the master's priority. Its inaccuracy amounts up to 35%. It is interesting to note, that the TLM (b), althought it does not perform any content resolution, exhibits a smaller error than the TLM (a).

The measurements show very similar results for the low priority master. Hence, its graph is omitted for brevity, but can be found in [11].

Additionally to the just analyzed application latency, the application finish time is of interest for design decisions. Therefore, we have evaluated the same experimental data in terms of the cumulative transfer time, which is the sum of all user transaction durations. Figure 7 shows the error in the cumulative duration for the high priority master using

locked transfers. The graph for the low priority master is omitted, since it is very similar.

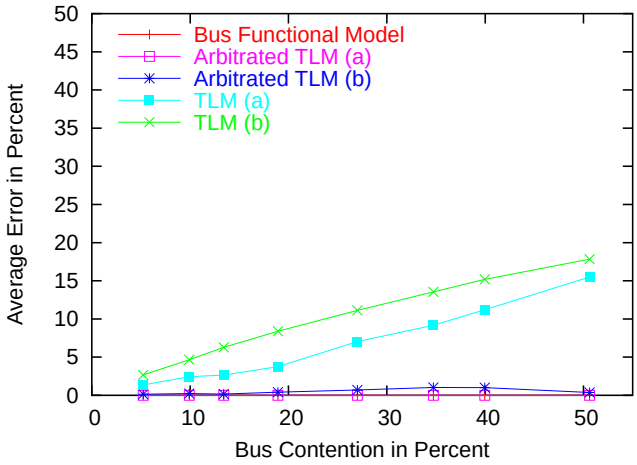


Figure 7. Locked accuracy by cumulative duration.

As in the previous graph, the lines for both BFM and ATLM (a) lie on top of the x-axis. The graph reveals that the mispredictions made by the less accurate ATLM (b) do average out. Both ATLM variants are good predictions for the application finish time. Both TLM variants exhibit a greater error. The TLM (b)'s error is now larger than TLM (a)'s, since the (b) variance constantly predicts a too short duration (due to the lack of arbitration). Its errors accumulate and do not average out. This makes the TLM (a) a better choice with an error of at most 15% for 50% bus contention.

5.2.3 Analysis for Unlocked Transfers

Additionally to the previous analyzed locked transfers we have repeated the same experiment for unlocked transfers. Here a burst may be preempted by a higher priority bus master, and has to be resumed later. As before, we analyzed the accuracy for both: the individual transfer duration and the cumulative transfer time. Both analysis aspects yield similar results, therefore only one - the accuracy based on cumulative transfer time - is shown in Figure 8. Since the results differ by priority, the graphs for the high priority master and the low priority master are shown side by side.

Figure 8 shows that only the BFM yields accurate results. With unlocked transfers, an arbitration decision is also made within a bus transaction. Therefore the ATLM (a) does now show an error of up to 35% for the low priority master. It handles arbitration only at the bus transaction granularity. The difference between the variances of the ATLM becomes insignificant. The ATLMs perform similar for the high and the low priority master.

The two TLM variances yield opposite results for the different masters. For the high priority master, TLM (b) is the more accurate choice. With its lack of arbitration it models an always available bus. This is close to the reality for the high priority master. A request can preempt the lower priority master and bus ownership is obtained within a few bus cycles. For the low priority master on the other hand, the overly optimistic TLM (b) shows a linear increasing error of up to 50% for 50% bus contention. This master gets increasingly often preempted. The dramatic difference in accuracy between the masters makes the TLM (b) not a viable solution. The TLM (a), which models contention resolution with a semaphore per user transaction, performs more consistent, but can only give a rough timing estimate. It exhibits a linear increasing error of up to 45% for the high priority master; the test requires arbitration handling for each bus cycle.

Combining the analysis results for the locked and the unlocked transfers provides a ground for selecting between the variances of TLM and ATLM. Between the TLM variations, the TLM (a) is selected. It exhibited smaller errors in the cumulative tests for the locked transfers, and was more consistent in its predictions for the unlocked transfers. The ATLM (a) is chosen among the ATLM variations, since it was accurate in the locked transfer tests and both variations performed similarly for the unlocked transfers.

In general, the expectations have been confirmed. The more abstract models simulate much faster, but also deliver less accurate results. However, the results are strongly correlated with the application characteristics. Some guidelines, extracted from this correlation, are described next.

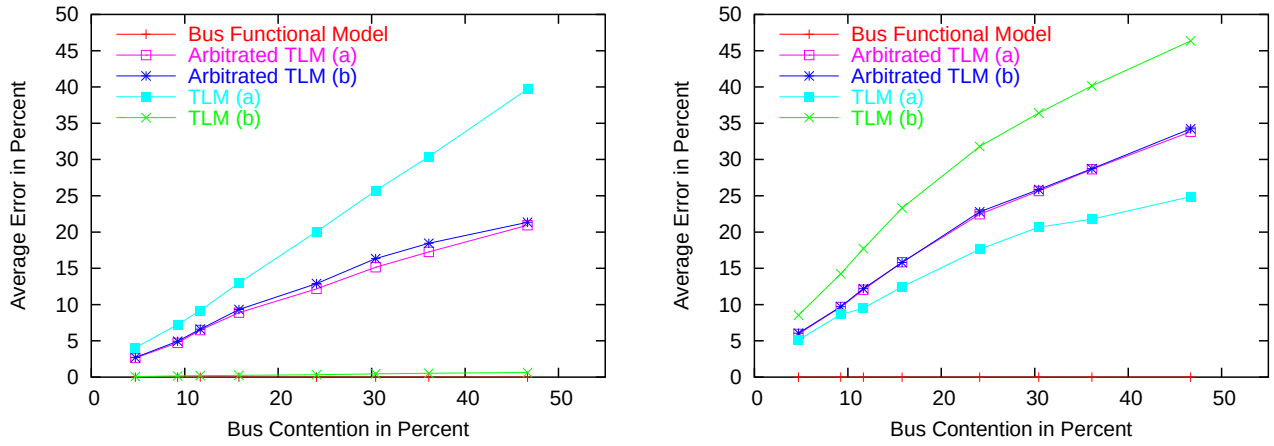
6. Conclusion

In this paper we reported on a case study for abstract communication modeling using the AMBA AHB. We implemented three major models: the bus functional model (BFM), the arbitrated transaction level model (ATLM) and the transaction level model (TLM). Additionally, we created two variances for each the ATLM and the TLM.

We evaluated usability of the models. For one, we measured a speedup in simulation performance of two magnitudes per major model. However, the variances within the ALTM and the TLM executed with a very similar speed. It is mainly the granularity of data and arbitration handling, that contributes to the speedup.

We analyzed in detail each model's simulation accuracy and selected one of the variances for each the TLM and the ATLM. Based on the analysis outcome, Table 2 lists the fastest model, that yields acceptable results for a given environment and simulation focus.

For computation bound applications, or when almost no bus contention is expected, all models yield close to accurate results. The very abstract TLM delivers acceptable results the fastest. Otherwise, the TLM is only of very lim-



(a) high priority master

(b) low priority master

Figure 8. Unlocked accuracy by cumulative duration.

Environment Condition	Model	Speedup
• single master bus • no bus contention	TLM	10^4
• only locked transfers • unlocked transfers, low contention	ATLM	10^2
• unlocked transfers, high contention	BFM	10^0

Table 2. Conclusion summary

ited use for simulating high contention scenarios. Its error reached 45% for unlocked transfers.

A system that only uses locked transfers can be accurately simulated by the ATLM. Its results are also acceptable for unlocked transfers in systems with a low bus contention. However, when simulating unlocked transfers under high bus contention, the ATLM exhibits an error of up to 35%. In this case, only the BFM yields accurate results.

We presented in this paper a classification of the analyzed models for their use in typical modeling situations. This classification enables the designer to select the proper model and achieve fast and accurate simulation of communication. As an indication for the generality of the presented classification, it has to be noted, that we observed similar results when modeling a very different (serial, off-chip) protocol [10].

References

[1] Advanced RISC Machines Ltd (ARM). AMBA AHB Cycle Level Interface (AHB CLI) Specification, AHBCLI.1.1.0. www.arm.com/products/solutions/ahbcli.html.

[2] Advanced RISC Machines Ltd (ARM). AMBA Specification (Revision 2.0), ARM IHI 0011A. www.arm.com/products/solutions/AMBA_Spec.html.

[3] Marco Caldari et al. Transaction-Level Models for AMBA Bus Architecture Using SystemC 2.0. In *DATE*, Munich, Germany, March 2003.

[4] M. Coppola et al. IPSIM: SystemC 3.0 Enhancements for Communication Refinement. In *DATE*, Munich, Germany, March 2003.

[5] D. Gajski, J. Zhu, R. Dömer, A. Gerstlauer, and S. Zhao. *SpecC: Specification Language and Design Methodology*. Kluwer Academic Publishers, 2000.

[6] A. Gerstlauer et al. System-Level Communication Modeling for Network-on-Chip Synthesis. In *ASPDAC*, Shanghai, China, January 2005.

[7] T. Grötker, S. Liao, G. Martin, and S. Swan. *System Design with SystemC*. Kluwer Academic Publishers, 2002.

[8] Internation Organization for Standardization (ISO). *Reference Model of Open System Interconnection (OSI)*, second edition, 1994. ISO/IEC 7498 Standard.

[9] S. Pasricha et al. Fast Exploration of Bus-based On-chip Communication Architectures. In *CODES and ISSS*, Stockholm, Sweden, September 2004.

[10] G. Schirner and R. Dömer. Abstract Communication Modeling: A Case Study Using the CAN Automotive Bus. In A. Rettberg, M. Zanella, and F. Rammig, editors, *From Specification to Embedded Systems Application*, Manaus, Brazil, August 2005. Springer.

[11] G. Schirner and R. Dömer. System Level Modeling of an AMBA Bus. Technical Report CECS-TR-05-03, Center for Embedded Computer Systems, University of California, Irvine, March 2005.

[12] M. Sgroi et al. Addressing the System-on-a-Chip interconnect woes through communication based design. In *DAC*, June 2001.

[13] R. Siegmund and D. Müller. SystemC^{SV}: An extension of SystemC for mixed multi-level communication modeling and interface-based system design. In *DATE*, Munich, Germany, March 2001.