

System Level Architecture Evaluation and Optimization: an Industrial Case Study with AMBA3 AXI

Jong-Eun Lee*, Woo-Cheol Kwon*, Tae-Hun Kim*, Eui-Young Chung*,
Kyu-Myung Choi*, Jeong-Taek Kong*, Soo-Kwan Eo*, and David Gwilt**

Abstract—This paper presents a system level architecture evaluation technique that leverages transaction level modeling but also significantly extends it to the realm of system level performance evaluation. A major issue lies with the modeling effort. To reduce the modeling effort the proposed technique develops the concept of worst case scenarios. Since the memory controller is often found to be an important component that critically affects the system performance and thus needs optimization, the paper further addresses how to evaluate and optimize the memory controllers, focusing on the test environment and the methodology. The paper also presents an industrial case study using a real state-of-the-art design. In the case study, it is reported that the proposed technique has helped successfully find the performance bottleneck and provide appropriate feedback on time.

Index Terms—Transaction level modeling, worst case scenario, mobile application processor, bus interconnect, memory controller, AMBA3 AXI.

I. INTRODUCTION

A Mobile Application Processor (MAP), which extends a RISC processor with hardware IPs, provides a

cost-effective, low power, and high performance solution to mobile and general purpose applications [1, 2]. One such MAP announced by Samsung is targeted for enabling diverse multimedia content in 3G mobile handheld devices such as smart phones and PDAs [1]. To support seamless real time video images and high-density multimedia services the MAP contains up to 64 Kbytes of cache memory. Another example of MAP is the Nomadik architecture [2] from STMicroelectronics supporting various multimedia functions including MP3, AAC, H.264, and MPEG4.

Due to the stringent cost and performance requirements as well as the shrinking time-to-market window, designing MAPs poses new challenges to system architects in the SOC era. One such challenge is to evaluate the system level architecture with high accuracy when there is a set of IPs (including busses and memory controllers) interacting with each other. Our approach is based on the Transaction Level Modeling (TLM) technique [3] but significantly enhances it, applying the technique at the true system level, not merely IP level. For the highest credibility of the system level evaluation results, we choose Bus Cycle Accurate (BCA) level of modeling among different flavors of TLM. While it provides the most reliable simulation results, BCA modeling suffers from slower simulation speed and, more importantly, higher modeling effort. Our system level evaluation technique tries to minimize the modeling effort by exploiting the worst case scenario concept. We develop the idea of worst case scenarios in the context of system level analysis and modeling, and also provide an industrial case study using a real state-of-the-art design. Our case study demonstrates the efficacy of our system level design methodology in analyzing and evaluating the system architectures for today's complex

Manuscript received October 20, 2005; revised December 3, 2005.

*SoC R&D Center, System LSI Samsung Electronics, Co. Giheung-eup, Yongin-si, Gyeonggi-do, Korea

**Fabric IP Division, ARM Ltd. 110 Fulbourn Road, Cambridge, CB1 9NJ, United Kingdom

E-mail : Jongeun.Lee@samsung.com

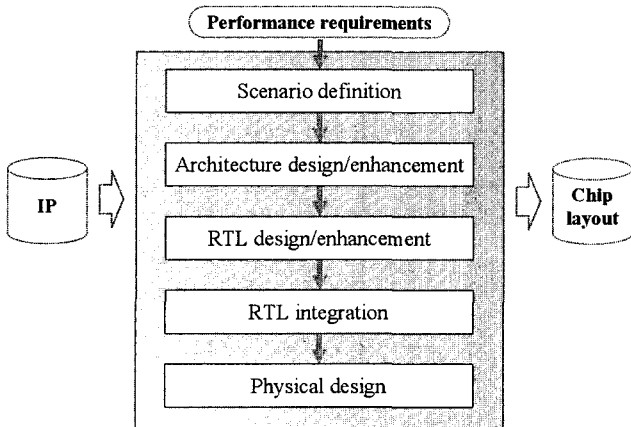


Fig. 1. Overall design flow from IP's to chip layout.

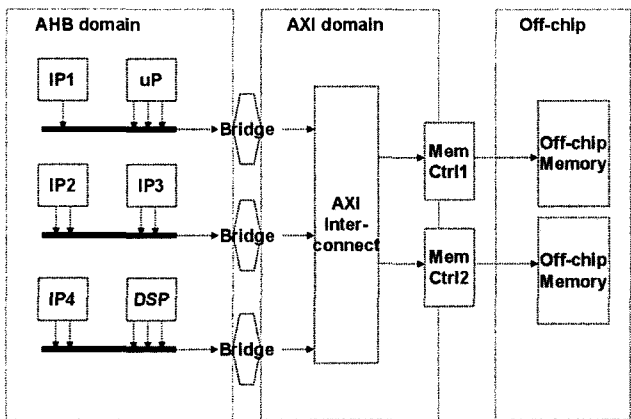


Fig. 2. Example system architecture.

Thus we further address how to evaluate and optimize the memory controllers, focusing on our test environment and methodology. Using our architecture evaluation schemes we could successfully find the performance bottleneck and provide appropriate feedback on time to designers and system integrators.

II. DESIGN FLOW

Figure 1 illustrates our overall design flow starting from a list of IPs to chip layout. The goal of the design process is to determine from the given set of IPs an interconnect and memory subsystem architecture so as to meet the performance requirements. Figure 2 illustrates an example system architecture, where the IPs can access the memory through three AHB bus layers, one AMBA3 AXI [4] interconnect, and the memory controllers. In the figure the interconnect architecture parameters may include the number of AHB layers, the layer organization (e.g., which IP is located in which layer), and the bus arbitration algorithm, while it is also conceivable to explore different bus protocols and different memory controller algorithms. We assume that IPs can be provided either as TLM models or as RTL netlists. In the case of the TLM models, IP design is still under way, so there may be more room for architectural exploration, whereas in the latter case there may be a more restricted design space to explore. Our methodology can be applied in both cases.

SOC designs. As a result of our system level analysis the memory controller was found to be the single most important component that affects the overall performance.

Table. 1. Worst case scenario example (MPEG4 decoding).

IP	Action description			Timing restriction	
	Action	Parameter	Data dependency	Throughput	Other constraint
MPEG Codec	Decode	Picture_size = CIF; input_stream = abc.mpg; f_core = 1;	Use stream data from DMA output	30 frame/s	None
Deblock Filter	Deblock	Size = CIF; DMA_size = 16;	Use MPEG Codec output	30 frame/s	None
Post Processor	Color space conversion	Input = 4:2:0 YCbCr, CIF; Output = RGB 16 bpp, CIF;	Use Deblock output	30 frame/s	None
LCD Controller	LCD control	Size = CIF; DMA_size = 4;	Use Post Processor output	60 Hz	Real-time constraint

III. SYSTEM LEVEL ARCHITECTURE EVALUATION

1. Worst Case Scenario

Our system level architecture evaluation process starts with *scenario definition*, where a scenario can be considered as a refined form of performance requirements. The performance requirements do not have to be detailed but should show what will be the final performance perceived by the end user. An example of the performance requirements is listed in Table 2. From the performance requirements, we derive the worst case scenarios.

A *scenario* is a set of actions played by participating IPs with (if any) timing restrictions specified. For example, the performance requirement for MPEG4 decoding states that several IPs including MPEG Codec, Deblock Filter, Post Processor, and LCD Controller should perform their operations at certain rates. The MPEG4 decoding scenario can be described as in Table 1. Note that the *action description* part represents the test condition that needs to be modeled, and the *timing restriction* part represents the targets that should be matched by the simulation results.

The action description may have “modes” (or parameters) in which each IP performs its task. For instance, the MPEG Codec is in the “decode” mode in the MPEG4 decoding scenario. Some parameters are free to change, however, either by user at run time or by designer at design time. These optional parameters produce a set of scenarios, all of which may be interesting. Preparing models and running simulations for all of them is a certain waste of effort, and we prune most of the scenarios and select only the most interesting ones, those that demand the greatest performance to the system architecture (i.e., the bus and memory controller subsystem). Those are called worst case scenarios.

2. TLM Modeling and Simulation

Evaluating the system level architecture fast and accurately is key to finding optimal interconnect architectures of an SOC. For the most accurate analysis results, the dynamic effects between various IPs, busses, and memory controllers should be taken into account. Due

Table 2. Performance requirement example.

Category	Specification
Display	CIF, 16 bit/pixel
MPEG4 encoding	CIF, 15 frame/s
MPEG4 decoding	CIF, 30 frame/s
Video teleconferencing	QCIF, 15 frame/s
Camera	3 M pixel/s
3D acceleration	1 M triangle/s

to its prohibitively large computational resource requirement, RTL simulation is often not feasible for this kind of system level analysis. Also hardware emulation is too costly an approach at the early stages of the design process, since the system architecture is very likely to change as a result of the architecture evaluation and exploration.

We use the Transaction Level Modeling (TLM) technique to capture the behavior of IPs, busses, and memory controllers, and to evaluate and explore different system level architectures. For the most credible results, we model the components at the Bus Cycle Accurate (BCA) level of abstraction. This ensures that dynamic effects such as bus arbitration and memory controller scheduling are accurately reflected in our TLM simulation results. One big drawback of the bus cycle accurate modeling at the system level is the modeling effort. To minimize the modeling effort we model only the IPs that appear in the worst case scenarios. Modern MAPs consist of at least a few dozens of IPs to integrate at the top level. Modeling only a subset of the IPs is the most obvious way to reduce the modeling effort. Further, each IP has a number of modes and parameters. It is arguable that modeling and testing an IP for all its modes and parameters may require more effort than modeling multiple IPs for a subset of modes and parameters. Thus we model only those features that will be used in the worst case scenarios, which should reduce the modeling effort.

Figure 3 illustrates our system architecture evaluation flow using TLM. Using the worst case scenarios we can reduce the number of IPs and of simulation runs to minimum. The major effort is then put into (if IPs are provided in RTL only) developing TLM models and running simulations making sure all the action description

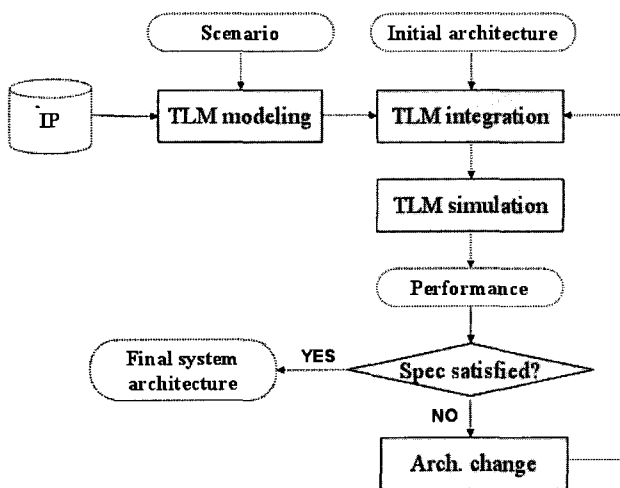


Fig. 3. System architecture evaluation flow.

part has been correctly taken into the simulation set-up. For instance, the data dependency condition can be enforced during the TLM integration step through the software on a microprocessor that correctly sequences the IP operations.

3. Architecture Evaluation Results

We have performed architecture evaluation for a set of system architectures for our latest design for a MAP. The architecture is similar to Figure 2, but has more AHB layers and much more IPs. We have used the PrimeCell AXI configurable interconnect [5] and the PrimeCell AXI memory controller [6], which had not been used before in any commercial products (not even by other companies). Therefore it is a real advantage to obtain such a system level architecture evaluation results before FPGA or hardware emulation results become available.

Out of the performance requirements that are similar to Table 2 we have derived four worst case scenarios, each of which is similar to Table 1. For each scenario, we have set

Table 3. Architecture exploration results.

	DMA_size = 4	DMA_size = 16
Original layer organization	LCD Controller misses real time constraint.	27% faster than required
Separate LCD Controller	23% faster than required	27% faster than required

up the TLM simulation environment, preparing the TLM models of necessary IPs. The data dependency requirement was implemented with the software on the ARM processor, which controls the execution of IPs. After performing TLM simulations, we validated our simulation results with the profiling data on each IP's memory access bandwidth.

Table 3 lists the architecture exploration results for one of the worst case scenarios. We tried four different system architectures, changing the layer organization (making a separate layer for the LCD Controller) and a system parameter (DMA_size of the LCD Controller). The original architecture couldn't meet the performance specification because the LCD Controller missed the real time constraint. First we changed the layer organization allocating a new layer to the LCD Controller. It worked but it was a costlier solution, compared to changing the system parameter, which was only discovered after many simulations.

IV. MEMORY CONTROLLER ENHANCEMENT

The system level architecture exploration has revealed that the memory controller is critical in delivering high performance at the system level. Thus we have focused our architecture enhancement on the memory controller. Figure 4 illustrates our design flow to refine the memory controller. At the heart of the refinement flow is a test environment, which is illustrated in Figure 5. With our automated test environment, which is built using Specman eVC [7], we could easily see if any modification in the RTL has unexpected side effects. Our test environment also has implemented *scoreboarding* and is flexible enough to support different data widths to the memory. The test environment is used not only for functional verification but also to make sure that the memory controller does not waste cycles unnecessarily, by counting the latency of typical transactions in various situations. With this methodology, we could find at least two cases where significant performance enhancement can be made.

1. Write Bank Interleaving

During the simulation of the initial RTL, it has been

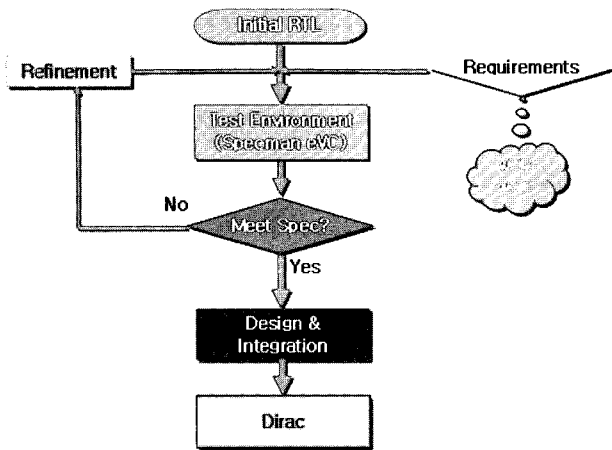


Fig. 4. Memory controller refinement flow.

found that the memory controller does not perform write bank interleaving even when it is significantly advantageous. Figure 6 shows the simulation results highlighting the ineffective behavior of the memory controller. The waveform shown is for a transaction with the burst size being four. In this particular case the four transfers access four different banks of the memory and there is no prior transaction that is still in execution. Then the ideal memory controller should issue row activation commands for each of the four memory banks and try to expedite the overall transaction, hiding the row activation latency. However as the figure suggests, the initial memory controller wasn't doing this obvious optimization. After identifying the cause of the problem we could enhance the write performance when write bank interleaving can be performed. Figure shows the transaction after write bank interleaving is implemented in the memory controller. In this case the write latency is reduced from 71 cycles to 32 cycles (corresponding to 45% improvement).

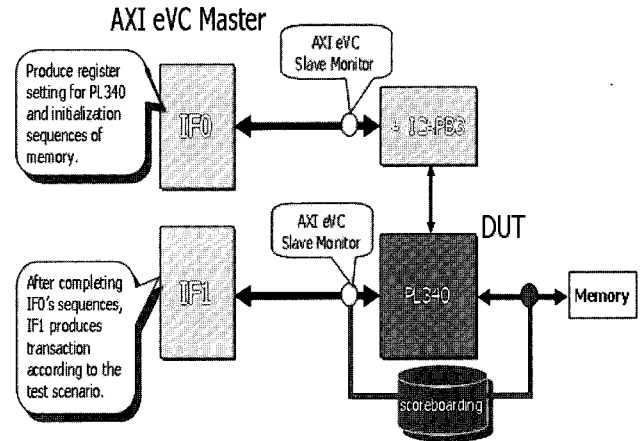


Fig. 5. Memory controller test environment.

2. Write Data FIFO Merging

Another example of improving the memory controller performance through our automatic test environment is related to the write data FIFO, which resides in the memory controller. To support multiple outstanding transactions the memory controller has an internal write data FIFO. But if the write data FIFO becomes full, a transaction with more than one write data items would enter the FIFO as if it were two separate transactions. We were able to identify this erroneous behavior using our test environment, as depicted in Figure 8. After removing this inefficiency by merging the (incorrectly separated) two transactions, the memory controller could perform as it should (reducing the latency from 17 to 13 cycles, in the example).

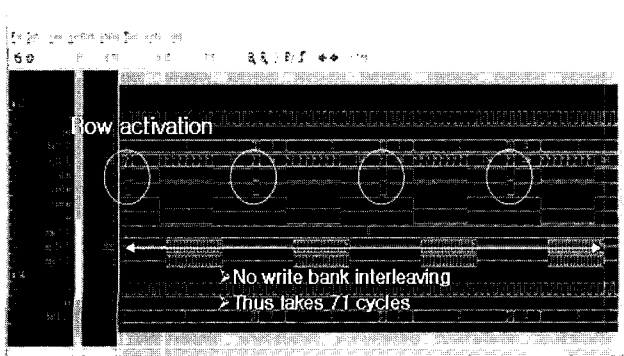


Fig. 6. Before write bank interleaving.

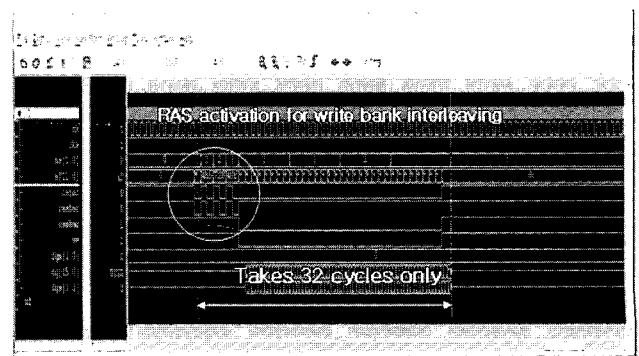


Fig. 7. After write bank interleaving.

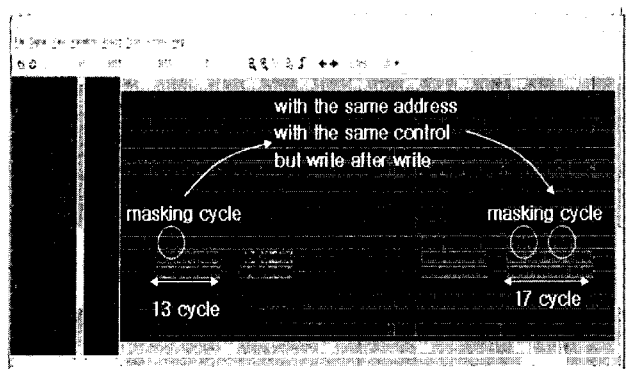


Fig. 8. Erroneous behavior when write data FIFO is full.

V. CONCLUSIONS

In this paper we presented a case study designing a complex SOC for modern mobile application processors, which typically have stringent requirements for cost, power, and performance. To address the challenge of analyzing and exploring different interconnect architectures in a timely manner with the highest accuracy, we used the TLM technique at the BCA modeling level. To reduce the high modeling effort of BCA modeling, we exploited the concept of worst case scenarios for modeling system level architectures. Our architecture analysis technique provides reliable system level performance estimates taking into account dynamic effects between IPs and busses, and also enables system level architecture exploration. We also addressed how to evaluate and optimize the memory controller, which was found to be a system level component with critical importance for MAP performance. We demonstrated the efficacy of our design evaluation technique using a real industrial design. Applying our architecture evaluation technique we could easily find performance bottlenecks and provide appropriate feedback to designers and system integrators.

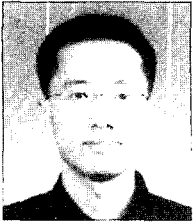
Our future work includes investigating and quantifying how much our technique can save the modeling effort for system level architecture evaluation. Being based on scenarios and ultimately on performance requirements, our system level evaluation technique is susceptible to the requirement change. An incremental method therefore would be desirable to accommodate the performance requirement change.

VI. ACKNOWLEDGMENTS

The authors would like to thank Dr. Byeong Min in CAE Center, Samsung and Harry Cho in Processor Architecture Lab., Samsung for their useful comments and discussion.

REFERENCES

- [1] "Mobile application processor for 3G phones," 3G Newsletter, Sept. 2004, online document available at <http://www.3g.co.uk/PR/Sept2004/8344.htm>, last accessed July 20, 2005.
- [2] "New multimedia application processor chips for 2.5/3G mobile phones," 3G Newsletter, Feb. 2003, online document available at <http://www.3g.co.uk/PR/Feb2003/4832.htm>, last accessed July 20, 2005.
- [3] T. Grotker et al., *System Design with SystemC*, Kluwer, 2002.
- [4] "AMBA AXI Protocol Specification," ARM Limited, 2003.
- [5] "PrimeCell AXI Configurable Interconnect," ARM Limited, Ref: ARM DDI 0354A, Dec 2004.
- [6] "ARM PrimeCell Dynamic Memory Controller Technical Reference Manual," ARM Limited, Ref: ARM DDI 0331A, June 2004.
- [7] Specman eVC, further information available on <http://www.cadence.com/verisity>.



Jong-Eun Lee He received a B.S. and an M.S. in electrical engineering and a Ph.D. in electrical engineering and computer science all from Seoul National University. He was a visiting scholar in Center for Embedded Computer Systems, University of California, Irvine during January 2002 through March 2003. He is currently with Design Technology Group of Samsung SoC R&D Center in Korea. His research interests include automation of embedded system design, reconfigurable processor architecture, and SoC on-chip interconnect.



Woo-Cheol Kwon He is a researcher in CAE center, Semiconductor Division, Samsung Electronics Co., Ltd in Korea. His research interests include system-level design, on-chip interconnects and algorithm design & analysis. He received the B.S. and M.S. degrees in computer science from Korea Advanced Institute of Science and Technology.



Tae-Hun Kim He is a researcher in CAE center, Semiconductor Division, Samsung Electronics Co., Ltd in Korea. His research interests include high-performance architecture system and high-bandwidth memory controller design. He received the B.S. degrees in electronic engineering from Inha university.



Eui-Young Chung He received the PhD degree in electrical engineering from Stanford University in 2002. He is an associate professor of electrical and electronic engineering at Yonsei University, Seoul, Korea. From 1990 to 2005, he was a principal engineer at SoC R&D center, Samsung Electronics, Kiheung, Korea. Dr. Chung's research interests are system architecture and VLSI design including all aspects of computer aided design with the special emphasis on low power applications and in the design of mobile systems.



Kyu-Myung Choi He received B.S. and M.S. degrees from Hanyang University, Seoul, Korea, in 1983 and 1985, respectively and Ph.D degree in electrical engineering from the University of Pittsburgh, USA, in 1995. His Ph.D research focused on the development of algorithms for CAD tools on system-level design automation. Since 1985, he is with Samsung Electronics as an engineer of CAE(Computer Aided Engineering) team. He studied his Ph.D course as a Samsung scholarship. Now he is a Vice President of CAE center, System-LSI Division, Samsung Electronics.



Jeong-Taek Kong He received the B.S. degree in electronics engineering from Hanyang University, Korea, in 1981, the M.S. degree in electronics engineering from Yonsei University, Korea, in 1983, and the Ph.D. degree in electrical engineering from Duke University, Durham, NC, in 1994. From 1983 to 1990, he was with Samsung Electronics Co., Ltd., as a VLSI CAD manager. From 1990 to 1994, he was at Duke University granted by a Fellowship from Samsung Electronics Co., Ltd. Currently, he is with Semiconductor Business, Samsung Electronics Co., as VP of CAE Team. He has authored and coauthored more than 110 technical papers in international journals and conferences and coauthored a book titled Digital Timing Macromodeling for VLSI Design Verification (Norwell, MA: Kluwer, 1995). His research interests focus on various VLSI CAD tools and design technologies.

He has served on the program committees of IEEE International Workshop on Statistical Metrology, International Conference on VLSI and CAD, International Symposium on Quality of Electronic Design, International Conference on Simulation of Semiconductor Processes and Devices, International Workshop on Behavioral Modeling and Simulation, International Symposium on Low Power Electronics and Design, International SOC Design Conference, and International Electron Devices Meeting. He is a Member of Nanoelectronics and Giga-Scale Systems (NaGS) Technical Committee and serves as a Distinguished Lecturer for the IEEE Circuits and Systems Society. He was an Associate Editor of IEEE Transactions on Circuits and Systems-II and IEEE

Transactions on Very Large Scale Integrated (VLSI) Systems and was nominated as a member of IMEC Scientific Advisory Board in Belgium in 2004.



Soo-Kwan Eo He has been senior vice president of Samsung Electronics' SoC R&D Center focusing on the ESL Design since late 2002. He currently directs the team that deals in SoC on-chip communications fabrications including NoC and pre-silicon SoC solution development methodology. His team developed the Samsung's Virtual Platform, called ViP, low power architecture design technology, HW/SW co-design technology, and Samsung SoC bus architecture. He played a key role to deploy these ESL design technologies within companies. Mr. Eo received MS in ECE from the University of Arizona Tucson, Arizona in 1986 and had worked in various companies including Cadence, Synopsys, Intel, VIA Technologies in San Jose, California for 16 years before he joined the Samsung.



Dave Gwilt He is the engineering manager for the Fabric IP Business Unit of ARM in Cambridge. His industrial experience includes development of the ARM920T and ARM1136JS processors, and architecture of the PL340 family of SDRAM controllers and PL330 family of DMA controllers. His technology interests include transaction-level modeling of IP. He received a Masters degree in engineering from Cambridge University.