Glossary

A

AHB advanced high-performance bus. AHB is an AMBA high speed bus.

ALU arithmetic/logic unit. The arithmetic/logic unit is a fundamental building block of the CPU's internal architecture. It performs logical and arithmetic operations.

AMBA The AMBA bus is a an open bus standard promulgated by ARM.

APB advanced peripheral bus. APB is an AMBA bus. The APB bus is devoted to low-speed peripherals and it is optimized to minimize power consumption and to reduce interface complexity.

API application programming interface. An application programming interface is a set of declarations of the functions that an operating system, library, or service provides to support requests made by computer programs.

ASB advanced system bus. ASB is an AMBA bus which is mainly deprecated and it has been substituted by AHB.

ASIC application-specific integrated circuit. An application-specific integrated circuit is an integrated circuit customized for a particular use, rather than intended for general-purpose use.

ASIP application-specific instruction set processor. An application-specific instruction set processor is a stored-memory CPU whose architecture is tailored for a particular set of applications.

AVC advanced video codec. Advanced video codec (AVC) is a standard for video compression. It is also known as H.264.

B

BDF Boolean dataflow. BDF is a model of computation based on dataflow. It is a generalization of SDF that sometimes yields to deadlock or boundedness analysis.

K. Popovici et al., *Embedded Software Design and Programming of Multiprocessor System-on-Chip*, Embedded Systems, DOI 10.1007/978-1-4419-5567-8,

BFM bus functional model. A bus functional model is used to simulate operating systems on the host machine. It transforms the functional memory access into hardware memory access.

BSB board support package. A board support package includes a set of minimal services necessary to load an operating system in an embedded system and the device drivers for all the devices on the board.

C

CABAC context adaptive binary arithmetic coding. CABAC is an entropy encoding algorithm, part of the H.264 video compression standard.

CAVLC context adaptive variable length coder. CAVLC is an entropy encoding algorithm, part of the H.264 video compression standard.

CORBA common object request broker architecture. The common object request broker architecture allows to execute different application objects that reside either in the same shared address space or remote address space. The application objects are described using interface definition languages (IDL).

Cycle accurate. cycle-accurate level is the most detailed abstraction level in our multiple abstraction levels design space exploration flow.

CPU central processing unit. A central processing unit is a description of a class of logic machines that can execute computer programs.

D

DCT discrete cosine transform. The DCT is a frequency transform used in video and image processing, whose coefficients describe the spatial frequency content of an image or video frame. The DCT operates on 2D set of pixels, in contrast to the Fourier transform which operates on a 1D signal.

DDF dynamic dataflow. DDF is a model of computation based on dataflow, which uses only runtime analysis to detect deadlock and boundedness.

DFT discrete Fourier transform. The DFT, occasionally called the finite Fourier transform, is a transform for Fourier analysis of finite-domain discrete-time signals. It expresses an input function in terms of a sum of sinusoidal components by determining the amplitude and phase of each component.

DMA direct memory access. The DMA is a component of the hardware architecture which allows accessing the memory independently of the processor. Usually it is used to initiate a data transfer between a local and global memory.

DSP digital signal processor. A digital signal processor is a specialized microprocessor designed specifically for digital signal processing, generally in real-time computing.

F

FIFO first in–first out. First in–first out is an abstraction in ways of organizing and manipulation of data relative to time and prioritization.

FPGA field programmable gate array. An FPGA is an integrated circuit which contains programmable logic components, which can be configured to perform complex combinational functions or logic operations. It can contain also memory elements.

FSM finite state machine. The FSM is a model of computation comprised of a finite number of states, transitions between those states and actions.

G

GPP general-purpose processor. A general-purpose processor is a processor that is not tied to, or integrated with, a particular language or piece of software.

H

HAL hardware abstraction layer. A hardware abstraction layer is an abstraction layer, implemented in software, between the physical hardware of a computer and the software that runs on that computer. Its role is to hide differences in hardware from most of the operating system kernel, so that most of the kernel-mode code does not need to be changed to run on systems with different hardware.

HdS hardware-dependent software. Hardware-dependent software is the part of an operating system which varies across microprocessor boards and is comprised notably of device drivers and boot code which performs hardware initialization.

I

IDCT inverse discrete cosine transform. A inverse discrete cosine transform expresses the opposite process of transforming a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies.

ILP instruction-level parallelism. Instruction-level parallelism is a measure of how many of the operations in a computer program can be performed simultaneously.

IQ inverse quantization. The IQ is used in multimedia decoding applications and represents the opposite of the quantization process. It consists of the multiplication of each of the 64 DCT coefficients by its corresponding quanta step size. The quanta steps are stored in the quantification tables.

ISR interrupt service routine. An interrupt service routine is a callback subroutine in an operating system or device driver whose execution is triggered by the reception of an interrupt.

ISS instruction set simulator. An instruction set simulator is a simulation model which mimics the behavior of a mainframe or microprocessor by “reading” instructions and maintaining internal variables which represent the processor’s registers.

ITRS international technology roadmap for semiconductors. The international technology roadmap for semiconductors, known throughout the world as the ITRS, is the 15-year assessment of the semiconductor industry's future technology requirements. The future needs drive present-day strategies for world-wide research and development among manufacturers' research facilities, universities, and national labs.

J

JPEG Joint Photographic Experts Group. JPEG is a commonly used method and standard of compression for photographic images and is created by the joint photographic experts group.

M

MCU microcontroller. The microcontroller is a type of CPU specialized mostly on control functions and combined with support functions such as timers, watchdog, serial and analog I/Os.

MIMD multiple instructions multiple data. Multiple instructions multiple data are a technique employed to achieve thread-level parallelism.

MIPS millions of instructions per second. MIPS stands for "millions of instructions per second" and is a rough measure of the performance of a CPU.

MJPEG motion JPEG. In multimedia, motion JPEG is an informal name for multimedia formats where each video frame or interlaced field of a digital video sequence is separately compressed as a JPEG image.

MPEG Moving Picture Experts Group. MPEG is a family of standards for video encoding and decoding.

MPI message-passing interface. MPI is a standard library for message passing that combines portability with high performance.

MPSoC multi-processor system-on-chip. The multi-processor system-on-chip is a system-on-chip which uses multiple processors, usually targeted for embedded applications.

N

NI network interface. The network interface is a component of a network-on-chip. It is responsible for providing *send/receive* operations for communicating subsystems, encapsulating these requests in packets, capturing and interpreting packets arriving from the NoC, and delivering them to the subsystems.

NoC network-on-chip. The network-on-chip is an interconnection component often used in MPSoC architectures to replace buses. It has many advantages compared to buses, which include high bandwidth, scalability, and power efficiency.

O

OS operating system. An operating system is the software component of a computer system that is responsible for the management and coordination of activities and sharing of the resources of the computer.

P

PIC programmable interrupt controller. The interrupt controller is a hardware component which handles external interrupts by according priorities to cope with external events. Its registers can be programmed by the designer to assign different priorities to different interrupt sources.

R

RAM random access memory. RAM represents a type of memory which can be accessed for read and write operations in any order.

RDL register description languages. The register description language allows specifying and implementing software-accessible hardware registers and memories.

RISC reduced instruction set computer. Reduced instruction set computer represents a CPU design strategy emphasizing the insight that simplified instructions which can be executed very quickly to provide higher performance.

ROM read-only memory. ROM represents a type of memory which can be only accessed for read operations. The data store in a ROM memory cannot be modified.

RTL register transfer level. In an integrated circuit design, register transfer level description is a way of describing the operation of a synchronous digital circuit. In RTL design, a circuit's behavior is defined in terms of the flow of signals (or transfer of data) between hardware registers and the logical operations performed on those signals.

Q

QoS quality of service. Quality of service is a measure of reliability used for data transmission over a shared network.

S

SA system architecture. The system architecture level is the highest abstraction level used in the software design for an MPSoC architecture. It captures both application and mapping specification.

SAD sum of absolute difference. The sum of absolute difference is a widely used, extremely simple video quality metric used for macroblock matching in motion estimation for video compression. It works by taking the absolute value of the difference between each pixel in the original macroblock and the corresponding pixel

in the macroblock of the reference frame, which is used for comparison. These differences are summed to create a simple metric of macroblock similarity.

SDF synchronous dataflow. SDF is a model of computation based on dataflow, which detects deadlock and boundedness.

SDG service dependency graph. The service dependency graph represents a unified model to capture the hardware/software interfaces for MPSoC architectures.

SIMD single instruction multiple data. The single instruction multiple data is a technique employed to achieve data-level parallelism, as in a vector processor.

SMP symmetric multi-processing. Symmetric multi-processing involves a multi-processor computer architecture where two or more identical processors can connect to a single shared main memory.

SoC system-on-chip. System-on-chip refers to integrating all components of a computer or other electronic system into a single integrated circuit (chip).

T

TA transaction accurate. The transaction-accurate level is an abstraction level used to simulate an application running on an operating system.

TLM transaction-level modeling. The transaction-level modeling is a high-level approach of modeling digital systems, where details of the communication among the components are separated from the details of the implementation of functional units or of the communication architecture.

TLP thread-level parallelism. The thread-level parallelism is a form of parallelization of computer code across multiple processors in parallel computing environments. It focuses on distributing the execution processes (threads) across different parallel computing nodes.

V

VA virtual architecture. The virtual architecture level is a more detailed abstraction level compared with the system architecture level. It serves to execute and debug a multi-threaded application.

VCI virtual component interface. Virtual component interface is a standard defined by the Virtual Socket Interface Alliance (VSIA). The overall objective is to obtain a general interface, such that intellectual property (IP), in the shape of virtual components (VCs) of any origin, can be connected to systems on chips of any chip integrator.

VLIW very long instruction word. VLIW is a style of computer architecture that issues multiple instructions or operations per clock cycle, but relies on static scheduling to determine the set of operations that can be performed concurrently.

VP virtual prototype. The virtual prototype level is a cycle-accurate abstraction level used to model hardware platforms. It serves to execute and debug the binary image of a multi-threaded application.

References

1. A386 library, http://a386.nocrew.org/
2. Aho, A. V., Lam, M. S., Sethi, R. and Ullman, J. D. Compilers: Principles, Techniques, and Tools (Second Edition). Addison Wesley, New York, August 2006.
3. HAL, http://www.altera.com/literature/hb/nios2/n2sw_nii5v2_02.pdf
4. AM2000 Processor Array Family, http://www.ambric.com
5. Anderson, A. J. Foundations of Computing Technology. CRC Press, Boca Raton, 1994, ISBN 0412598108
6. Technical documentation of ARM7 and ARM9 processors, AMBA Bus Technical Specification, RealView Compilation Tools Linker and Utilities Guide, Embedded Software Development with ADS v1.2, http://www.arm.com
7. Technical documentation of ARM MaxSim http://www.arm.com
8. Ascia, G., Catania, V. and Palesi, M. Mapping cores on network-on-chip. *International Journal of Computation Intelligence Research*, 1(2), 2005; 109–126.
9. mAgicV VLIW DSP and Diopsis http://www.atmelroma.it
10. Babcock, B., Babu, S., Datar, M., Motwani, P. and Widom, J. Models and issues in data stream systems. *Proceeding of 21st Symposium on Principles of Distributed Computing,* Monterey, California, USA, 2002, 1–16.
11. Bacivarov, I., Bouchhima, A., Yoo, S. and Jerraya, A. A. ChronoSym: A new approach for fast and accurate SoC cosimulation. *International Journal on Embedded Systems (IJES)*, 1(1), 2005; 103–111.
12. Bacivarov, I. Evaluation des performances pour les systèmes embarques heterogenes, multiprocesseurs monopuces. *Thèse de Doctorat INPG*, TIMA Laboratory, 2006 (in French)
13. Balasubramanian, K., Gokhale, A., Karsai, G., Sztipanovits, J. and Neema, S. Developing applications using model-driven design environments. *IEEE Computer Society*, 39(2), 2006; 33–40.
14. Barr, M. Memory types. *Embedded Systems Programming*, May 2001; 103–104.
15. Beltrame, G., Sciuto, D., Silvano, C., Paulin, P. and Bensoudane, E. An application mapping methodology and case study for multi-processor on-chip architectures. *Proceeding of VLSI-SoC 2006*, 16–18 October 2006, Nice, France, 146–151.
16. Benini, L. and De Micheli, G. Networks on chips: A new SoC paradigm. *IEEE Computer*, 35(1), 2002; 70–78.
17. Benini, L. and De Micheli, G. Networks on Chips: Technology and Tools. Morgan Kaufmann, San Francisco, 2006, ISBN-10:0-12-370521-5.
18. Berthet, C. Going mobile: The next horizon for multi-million gate designs in the semiconductor industry. *Proceeding of DAC*, 10–14 June 2002, New Orleans, USA, 375–378.
19. Berens, F. Algorithm to system-on-chip design flow that leverages system studio and systemC 2.0.1. *The Synopsys Verification Avenue Technical Bulletin* 4, 2, May 2004.
20. Bleuler, S., Laumanns, M., Thiele, L. and Zitzler, E. PISA- A platform and programming language independent interface for search algorithms. *Evolutionary Multi-Criterion Optimization (EMO 2003)*, Volume 2632/2003 of LNCS, Springer, 494–508.

21. Bonaciu, M. Plateforme flexible pour l'exploitation d'algorithmes et d'architectures en vue de réalisation d'application vidéo haute définition sur des architectures multiprocesseurs mono puces. *Thèse de Doctorat INPG*, TIMA Laboratory, 2006 (in French)
22. Bonaciu, M., Bouchhima, A., Youssef, W., Chen, X., Cesario, W. and Jerraya, A. High level architecture exploration for MPEG4 encoder with custom parameters. *Proceeding of ASP-DAC 2006*, January 2006, Yokohoma, Japan.
23. Bouchhima, A., Yoo, S. and Jerraya, A. A. Fast and accurate timed execution of high level embedded software using HW/SW interface simulation model. *Proceeding of ASP-DAC 2004*, January 2004, Yokohama, Japan, 469–474.
24. Bouchhima, A., Chen, X., Petrot, F., Cesario, W. O. and Jerraya, A. A. A Unified HW/SW interface model to remove discontinuities between HW and SW design. *Proceeding of EMSOFT'05*, 18–22 September 2005, New Jersey, USA, 159–163.
25. Bouchhima, A., Bacivarov, I., Youssef, W., Bonaciu, M. and Jerraya, A. A. Using abstract CPU subsystem simulation model for high level HW/SW architecture exploration. *Proceeding of ASP-DAC 2005*, 18–21 January 2005, Shanghai, China, 969–972.
26. Buck, J., Ha, S., Lee, E. and Messerschmitt, D. Ptolemy: A framework for simulating and prototyping heterogeneous systems. *International Journal of Computer Simulation*, 4, 1994; 155–182.
27. Busonera, G., Carta, S., Marongiu, A. and Raffo, L. Automatic Application Partitioning on FPGA/CPU Systems Based on Detailed Low-Level Information. *Proceeding of the 9th EUROMICRO Conference on Digital System Design*, 30 August – 1 September 2006, Croatia, 265–268.
28. Butenhof, D. R. Programming with POSIX Threads. Addison Wesley, New York, 1997, May, ISBN 0201633922.
29. Incisive Formal Verifier, http://www.cadence.com
30. Open Systems Glossary of Software Engineering Institute, Carnegie Mellon, http://www.sei.cmu.edu/opensystems/glossary.html
31. http://www.agilityds.com/news_and_events/press-release/dec3-2007.htm
32. Cesario, W. O., Lyonnard, D., Nicolescu, G., Paviot, Y., Yoo, S., Gauthier, L., Diaz-Nava, M. and Jerraya, A. A. Multiprocessor SoC platforms: A component-based design approach. *IEEE Design & Test of Computers*, 19(6), November-December 2002; 52–63.
33. Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D. and McDonald, J. Parallel Programming in OpenMP. *Morgan Kaufmann*, San Francisco, 2000; ISBN 9781558606715.October.
34. Chakraborty, S., Kunzli, S., Thiele, L., Herkersdorf, A. and Sagmeister, P. Performance evaluation of network processor architectures: Combining simulation with analytical estimation. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 41(5), April 2003; 641–665.
35. Cheong, E., Liebman, J., Liu, J. and Zhao, F. TinyGALS: A programming model for event-driven embedded systems. *Proceeding of 2003 ACM Symposium on Applied Computing*, Melbourne, Florida, USA, March 2003, 698–704.
36. Chen, K., Sztipanovits, J. and Neema, S. Toward a semantic anchoring infrastructure for domain-specific modeling languages. *Proceeding of EMSOFT 2005*, 19–22 September 2005, New Jersey, USA, 35–43.
37. Chen, J. W., Kao, C. Y. and Lin, Y. L. Introduction to H.264 advanced video coding. *Proceeding of ASP-DAC 2006*, 24–27 January 2006, Yokohama, Japan, 736–741.
38. Cho, Y., Yoo, S., Choi, K., Zergainoh, N. E. and Jerraya, A. A. Scheduler implementation in MPSoC design. *Proceeding of ASP-DAC 2005*, 18–21 January 2005, Shanghai, China, 151–156.
39. Cooley, J., Lewis, P. and Welch, P. The finite Fourier transform. *IEEE Transaction on Audio Electroacoustics*, 17(2), 1969; 77–85.
40. Coppola, M., Locatelli, R., Maruccia, G., Pieralisi, L. and Scandurra, A. Spidergon: A novel on-chip communication network. *Proceeding of International Symposium on System-on-Chip*, 16–18 November 2004.
41. ConvergenSC. Coware Processor Designer, http://www.coware.com

42. Culler, D., Singh, J. P. and Gupta, A. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann, San Francisco, 1998; August.ISBN 1558603433.
43. Desoli, G. et al. A new facility for dynamic control of program execution: DELI. *Proceeding of EMSOFT 2002*, Grenoble, France
44. Diopsis D940, http://www.atmel.com
45. Socrates chip integration platform, http://www.duolog.com
46. eCos, http://www.ecos.sourceware.org/docs-1.3.1/ref/ecos-ref.b.html
47. Erbes, C., Pimentel, A. D., Thompson, M. and Polstra, S. A framework for system-level modeling and simulation of embedded systems architecture. *EURASIP Journal on Embedded Systems*, Volume 2007, Article ID 82123, June 2007.
48. Fei, Y. and Ha, N. K. Functional partitioning for low power distributed systems of systems-on-a-chip. *Proceeding of ASP-DAC 2002*, 7–11 January 2002, Bangalore, India
49. Flake, P. and Schirrmeister, F. MPSoC demands system level design automation. *EDA Tech Forum*, 4(1), March 2007; 10–11.
50. Flynn, M. Some computer organization and their effectiveness. *IEEE Transactions on Computers*, C-21(9), 1972; 948–960.
51. FreeRTOS, http://www.freertos.org
52. Freescale DSP, http://www.freescale.com
53. Furtber, S. B. and Bainbridge, J. Future trends in SoC interconnect. *Proceedings of 2005 International Symposium on System-on-Chip*, November 2005, 183–186.
54. Gajski, D. D. SpecC: Specification Language and Design Methodology. Kluwer, Dordrecht, 2000.
55. Gauthier, L., Yoo, S. and Jerraya, A. A. Automatic generation and targeting of application specific operating systems and embedded systems software. *IEEE TCAD Journal*, 20, Nr. 11, November 2001.
56. Gerin, P., Shen, H., Chureau, A., Bouchhima, A. and Jerraya, A. A. Flexible and executable hardware/software interface modeling for multiprocessor SoC design using SystemC. *Proceeding of ASP-DAC 2007*, 390–395.
57. Gerstlauer, A., Shin, D., Domer, R. and Gajski, D. D. System-level communication modelling for network-on-chip synthesis. *Proceeding of ASP-DAC 2005*, 18–21 January 2005, Shanghai, China, 45–48.
58. Gilliers, F., Kordon, F. and Regep, D. A model based development approach for distributed embedded systems. *Proceeding of RISSEF 2002*, 137–151, 2002.
59. Gipper, J. and Dingee, D. Navigating general purpose processor roadmap. *Embedded Computing Design*, 2007.
60. Glass, C. and Ni, L. The turn model for adaptive routing. *Journal of ACM*, 41(5), 1994; 278–287.
61. GNU tools and documentation, http://www.gnu.org
62. Grotker, T., Liao, S., Martin, G. and Swan, S. System Design with SystemC. Kluwer, Dordrecht, 2002, ISBN 1402070721
63. Guerin, X., Popovici, K., Youssef, W., Rousseau, F. and Jerraya, A. Flexible application software generation for heterogeneous multi-processor system-on-chip. *Proceeding of COMPSAC 2007*, 23–27 July 2007, Beijing, China
64. Ha, S., Lee, C., Yi, Y., Kwon, S. and Joo, Y. P. Hardware-software codesign of multimedia embedded systems: The PeaCE. *Proceeding of 12th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'06)*, 2006, 207–214.
65. Ha, S. Model-based programming environment of embedded software for MPSoC. *Proceeding of ASP-DAC'07*, 23–26 January 2007, Yokohama, Japan, 330–335.
66. Han, S. I. et al. Buffer memory optimization for video codec application modeled in Simulink. *Proceeding of DAC 2006*, San Francisco, USA, 689–694.
67. Hassan, M. A., Sakanushi, K., Takeuchi, Y. and Imai, M. RTK-spec TRON: A simulation model of an ITRON Based RTOS Kernel in SystemC. *Proceeding of DATE 2005*, 7–11 March 2005, Munich, Germany, 554–559.

68. Hassan, M. A. and Imai, M. A system level modeling methodology for RTOS centric embedded systems. *Proceeding of 14th IFIP VLSI-SoC*, PhD Forum Digest of Papers, 16–18 October 2006, Nice, France, 62–67.
69. Hennessy, J. L. and Patterson, D. A. Computer Architecture: A Quantitative Approach, Third Edition. Printed by Elsevier Science Pte Ltd, Amsterdam, 2003, ISBN 1558605967.
70. Hong, S., Yoo, S., Lee, S., Nam, H. J., Yoo, B. S., Hwang, J., Song, D., Kim, J., Kim, J., Jin, H. S., Choi, K. M., Kong, J. T. and Eo, S. Creation and utilization of a virtual platform for embedded software optimization: An industrial case study. *Proceeding of CODES+ISSS 2006*, Seoul, Korea, 235–240.
71. Hwang, H., Oh, T., Jung, H. and Ha, S. Conversion of reference C code to dataflow model: H.264 Encoder Case Study. *Proceeding of ASP-DAC 2006*, 24–27 January 2006, Yokohama, Japan, 152–157.
72. Kahn, G. The semantics of a simple language for parallel programming. *Information Processing*, 1974; 471–475.
73. Kangas, T., Kukkala, P., Orsila, H., Salminen, E., Hannikainen, M., Hamalainen, T. D., Riihimaki, J. and Kuusilinna, K. UML-based multiprocessor SoC design framework. *ACM Transactions on Embedded Computing Systems (TECS)*, 5(2), 2006, 281–320.
74. Kempf, T., Doerper, M., Leupers, R., Ascheid, G., Meyr, H., Kogel, T. and Vanthournout, B. A modular simulation framework for spatial and temporal task mapping onto multi-processor SoC platforms. *Proceeding of DATE 2005*, 7–11 March 2005, Munich, Germany.
75. Kienhuis, B., Deprettere, Ed. F., van der Wolf, P. and Vissers, K. A. A methodology to design programmable embedded systems- the Y-Chart approach. *Lectures Notes in Computer Science, Volume 2268, Embedded Processor Design Challenges: Systems, Architectures, Modeling, and Simulation–SAMOS 2002*, Springer, 18–37.
76. Klingauf, W., Gadke, H. and Gunzel, R. TRAIN: A virtual transaction layer architecture for TLM-based HW/SW codesign of synthesizable MPSoC. *Proceeding of DATE 2006*, 6–10 March 2006, Munich, Germany, 1318–1323.
77. Klingauf, W., Gunzel, R. and Schroder, C. Embedded software development on top of transaction-level models. *Proceeding of CODES+ISSS 2007*, 30 September–3 October 2007, Salzburg, Austria, 27–32.
78. de Kock, E. A. et al. Yapi: Application modeling for signal processing systems. *Proceeding of DAC 2000*, USA, 402–405.
79. Kogel, T., Wieferink, A., Meyr, H. and Kroll, A. SystemC based architecture exploration of a 3D graphic processor. *Proceeding of IEEE Workshop on Signal Processing Systems*, 26–28 September 2001, Antwerp, Belgium, 169–176.
80. Kramer, S., Gao, L., Weinstock, J., Leupers, R., Ascheid, G. and Meyr, H. HySim: A Fast Simulation Framework for Embedded Software Development. *Proceeding of CODES+ISSS 2007*, 30 September–5 October 2007, Salzburg, Austria
81. OpenCL standard, http://www.khronos.org/opencl/
82. Kunzli, S., Poletti, F., Benini, L. and Thiele, L. Combining simulation and formal methods for system-level performance analysis. *Proceeding of DATE 2006*, 6–10 March 2006, Munich, Germany, 236–241.
83. Kwon, S., Jung, H. and Ha, S. H.264 decoder algorithm specification and simulation in Simulink and PeaCE. *Proceeding of ISOCC 2004*, Seoul, Korea
84. NicheStack TCP/IP protocol stack, http://www.iniche.com/nichestack.php
85. Intel processors, http://www.intel.com
86. Jerraya, A. and Wolf, W. Hardware-software interface codesign for embedded systems. *Computer*, 38(2), February 2005; 63–69.
87. Jerraya, A., Bouchhima, A. and Petrot, F. Programming models and HW-SW Interfaces abstraction for Multi-Processor SoC. *Proceeding of DAC 2006*, San Francisco, USA, 280–285.
88. Lavenier, D. and Daumas, M. Architectures des ordinateurs. *Technique et Science Informatique*, 25(6), 2006.

89. Lee, E. A. Nesting and unnesting models of computation. *Presentation at the Seventh Biennial Ptolemy Miniconference*, 13 February 2007, Berkeley, CA, USA
90. Lieverse, P., Stefanov, T., van der Wolf, P. and Deprettere, E. System level design with SPADE: an M-JPEG case study. *Proceeding of ICCAD 2001*, 4–8 November 2001, San Jose, USA, 31–38.
91. Liu, J., Lajolo, M. and Sangiovanni-Vincentelli, A. Software timing analysis using HW/SW cosimulation and instruction set simulator. *Proceeding of the 6th International Workshop on Hardware/Software Co-design CODES/CASHE'98*, 15–18 March 1998, Seattle, Washington, 65–69.
92. LLVM Compiler Infrastructure, http://llvm.org/
93. LynxOS, http://www.lynuxworks.com/rtos/
94. Matlab and Simulink, The MathWorks Inc., http://www.mathworks.com
95. Magee, D. P. Matlab extensions for the development, testing and verification of real-time DSP software. *Proceeding of DAC 2005*, Anaheim, USA
96. Martin, G. Overview of the MPSoC design challenge. *Proceeding of DAC 2006*, 24–28 July 2006, San Francisco, USA, 274–279.
97. Mattioli, J., Museux, N., Jourdan, J., Saveant, P. and de Givry, S. A constraint optimization framework for mapping a digital signal processing application onto a parallel architecture. *Proceeding of the 7th International Conference on Principles and Practice of Constraint Programming*, 2001, 701–715.
98. http://www.multicore-association.org/press/080401.htm
99. Meijer, S., Walters, J., Snuijf, D. and Kienhuis, B. Automatic partitioning and mapping of stream-based applications onto the Intel IXP Network Processor. *Proceeding of Workshop on Software & Compilers for Embedded Systems (SCOPES'07)*, Nice, 20 April 2007.
100. CodeWarrior Development tools, http://www.metrowerks.com
101. Meyr, H. Application Specific Processors (ASIP): On design and implementation Efficiency. *Proceeding of SASIM 2006*, Nagoya, Japan
102. de Micheli, G., Ernst, R. and Wolf, W. Readings in Hardware/Software Co-design. Morgan Kaufmann, 2002, ISBN 1558607021.
103. Microchip Technologies Inc., http://www.microchip.com
104. Windows CE, http://www.microsoft.com/windows/embedded
105. Model driven architecture http://www.omg.org/mda/
106. Mohapatra, P. et al. Wormhole routing techniques for directly connected multicomputer systems. *ACM Computing Survey*, 30(3), 1998, 374–410
107. Moraes, F. et al. HERMES: An infrastructure for low area overhead packet-switching networks-on-chip integration. *VLSI Journal*, 38(1), 2004; 69–93.
108. MPI 2.2. Standard, http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf
109. MPICH, http://www.mcs.anl.gov/research/projects/mpi/mpich2/
110. Nicolescu, G. Specification et validation des systemes heterogenes embarques. *PhD Thesis*, TIMA Laboratory, 2002 (in French)
111. Nikolov, H., Stefanov, T. and Deprettere, E. Multi-processor system design with ESPAM. *Proceeding of CODES+ISSS'06*, 22–25 October 2006, Seoul, Korea, 211–216.
112. newlib, http://sourceware.org/newlib
113. Nexperia http://www.nxp.com
114. Nomadik, http://www.st.com
115. CUDA, http://www.nvidia.com/cuda
116. Object Management Group, CORBA Basics, 2006, http://www.omg.org/gettingstarted/corbafaq.htm
117. OpenMP specification for parallel programming, http://openmp.org/wp/
118. Open SystemC Initiative (OSCI) http://www.systemc.org
119. Oyamada, M., Wagner, F. R., Bonaciu, M., Cesario, W. and Jerraya, A. Software performance estimation in MPSoC design. *Proceeding of ASP-DAC'07*, 23–26 January 2007, Yokohama, Japan, 38–43.

120. Park, S., Olds, W., Shin, K. G. and Wang, S. Integrating virtual execution platform for accurate analysis in distributed real-time control system development. *Proceeding of RTSS 2007*, 3–6 December 2007, Tucson, Arizona, USA
121. Paolucci, P. S., Jerraya, A. A., Leupers, R., Thiele, L. and Vicini, P. SHAPES: a tiled scalable software hardware architecture platform for embedded systems. *Proceeding of CODES+ISSS 2006*, Seoul, Korea, 167–172.
122. Paulin, P., Pilkington, C., Langevin, M., Bensoudane, E., Lyonnard, D., Benny, O., Laviguer, B., Lo, D., Beltrame, G., Gagne, V. and Nicolescu, G. Parallel programming models for a multi-processor SoC platform applied to networking and multimedia. *IEEE Transactions on VLSI Journal*, 14(7), 2006; 667–680.
123. Pazos, N., Maxiaguine, A., Ienne, P. and Leblebici, Y. Parallel modelling paradigm in multimedia applications: Mapping and scheduling onto a multi-processor system-on-chip platform. *Proceedings of the International Global Signal Processing Conference*, Santa Clara, California, USA, September 2004.
124. Popovici, K. and Jerraya, A. A. Simulink based hardware-software codesign flow for heterogeneous MPSoC. *Proceeding of Summer Computer Simulation Conference (SCSC'07)*, 15–18 July 2007, San Diego, USA, 497–504.
125. Popovici, K. and Jerraya, A. Programming models for MPSoC. *Chapter 4 in Model Based Design of Heterogeneous Embedded Systems*. Ed. CRC Press, 2008.
126. Popovici, K., Guerin, X., Rousseau, F., Paolucci, P. S. and Jerraya, A. Platform based software design flow for heterogeneous MPSoC. *ACM Journal: Transactions on Embedded Computing Systems (TECS)*, 7(4), July 2008.
127. Pospiech, F. Hardware dependent Software (HdS). Multiprocessor SoC Aspects. An Introduction. *MPSoC 2003*, 7–11 July 2003, Chamonix, France
128. Pullini, A., Angiolini, F., Meloni, P., Atienza, D., Murali, S., Raffo, L., De Michelli, G. and Benini, L. NoC Design and Implementation in 65 nm Technology. *Proceeding of the 1st Internal Symposium on Networks-on-Chip*, 7–9 May 2007, Princeton, New Jersey, USA, 273–282.
129. Reyneri, L. M., Cucinotta, F., Serra, A. and Lavagno, L. A hardware-software codesign flow and IP library based on Simulink. *Proceeding of the 38th conference on Design Automation*, Las Vegas, United States, 2001, 593–598.
130. Richardson, I. and Sullivan, G. J. H264 and MPEG-4 Video Compression"
131. Rijpkema, E., Goossens, K. and Wielage, P. A router architecture for networks on silicon. *Proceeding of PROGRESS'01*, November 2001, 181–188.
132. Rijpkema, E., Goossens, K. and Radulescu, A. Tradeoffs in the design of a router with both guaranteed and best-effort services for networks on chip. *Proceeding of DATE'03*, March 2003, 350–355.
133. Roane, J. Electronic system level design for embedded systems. *EDA Tech Forum*, 4(1), March 2007; 14–16.
134. Rowson, J. A. Hardware/Software cosimulation. *Proceeding of DAC 1994,* San Diego, USA, 439–440.
135. RTLinux, http://www.fsmlabs.com
136. Qin, W., D'Errico, J. and Zhu, X. A multiprocessing approach to accelerate retargetable and portable dynamic-compiled instruction-set simulation. *Proceeding of CODES+ISSS 2006*, 22–25 October 2006, Seoul, Korea, 193–198.
137. Sarmento, A., Kriaa, L., Grasset, A., Youssef, W., Bouchhima, A., Rousseau, F., Cesario, W. and Jerraya, A. A. Service dependency graph, an efficient model for hardware/software interface modeling and generation for SoC design. *Proceeding of CODES-ISSS 2005*, New York, USA, 18–21 September 2005.
138. Schirner, G., Gertslauer, A. and Domer, R. Multifaced modeling of embedded processors for system level design (Abstract). *Proceeding of ASP-DAC 2007*, 23–26 January 2007, Yokohama, Japan, 384–389.

139. Schirner, G., Gertslauer, A. and Domer, R. Automatic generation of hardware dependent software for MPSoCs from abstract system specifications. *Proceeding of ASP-DAC 2008*, 21–24 January 2008, Seoul, Korea
140. Semeria, L. and Ghosh, A. Methodology for hardware/software co-verification in C/C++. *Proceeding of ASP-DAC 2000,* Yokohama, Japan, 405–408.
141. Shapes (Scalable Software Hardware Architecture Platform for Embedded Systems) European Project, http://shapes-p.org
142. Shin, D., Abdi, S. and Gajski, D. D. Automatic generation of bus functional models from transaction level models. *Proceeding of ASP-DAC 2004*, 27–30 January 2004, Yokohama, Japan, 756–758.
143. Shin, D., Gertslauer, A., Peng, J., Domer, R. and Gajski, D. D. Automatic generation of transaction-level models for rapid design space exploration. *Proceeding of CODES+ISSS 2006*, Seoul, Korea, 64–69.
144. Singhai, S., Ko, M. Y., Jinturkar, S., Moudgill, M. and Glossner, J. An Integrated ARM and Multi-core DSP Simulator. *Proceeding of CASES'07*, 30 September–3 October 2007, Salzburg, Austria, 33–37.
145. Skillicorn, D. and Talia, D. Models and languages for parallel computation. *ACM Computing Surveys*, 30(2), 1998; 123–169.
146. Spirit IP-XACT, http://www.spiritconsortium.com
147. Synopsys System Studio http://www.synopsys.com
148. Tanenbaum, A. S. Distributed Operating Systems. Prentice-Hall, Englewood Cliffs, 1995, ISBN 0132199084.
149. Tanenbaum, A. S. and Woodhull, A. S. Operating Systems: Design and Implementation. 1997, Prentice-Hall, Englewood Cliffs, ISBN 0136386776
150. Tanenbaum, A. S. Structured Computer Organization. 1999, Prentice-Hall, Englewood Cliffs, ISBN 013219901
151. Target Compiler Technologies, http://www.retarget.com
152. Xtensa processor architecture, XPRES Compiler, http://www.tensilica.com
153. Thies, W., Karczmarek, M. and Amarasinghe, S. StreamIt: a language for streaming applications. *Proceeding of International Conference on Compiler Construction*, April 2002, Grenoble, France
154. Thiele, L., Bacivarov, I., Haid, W. and Huang, K. Mapping applications to tiled multiprocessor systems. *Proceeding of Seventh International Conference on Application of Concurrency to System Design (ACSD 2007)*, 10–13 July 2007, Bratislava, Slovak Republic, 29–40.
155. Thompson, M., Nikolov, H., Stefanov, T., Pimentel, A. D., Erbas, C., Polstra, S. and Deprettere, E. F. A framework for rapid system-level exploration, synthesis, and programming of multimedia MP-SoCs. *Proceeding of CODES+ISSS 2007*, 30 September–3 October 2007, Salzburg, Austria, 9–14.
156. TI OMAP platform, DSPs, http://www.omap.com
157. Tile64 Processor Family, http://www.tilera.com
158. μITRON4.0 Specification, http://www.tron.org
159. Turley, J. Survey says: Software tools more important than chips. *Embedded Systems Design Journal*, 4-11-2005.
160. Van der Wolf, P. et al. Design and programming of embedded multiprocessors: An interface-centric approach. *Proceeding of CODES+ISSS 2004*, Stockholm, Sweden, 206–217.
161. Vanderperren, Y. and W. Dehaene. From UML/SysML to Matlab/Simulink: Current State and Future Perspectives. *Proceeding of Design Automation and Test in Europe, DATE 2006*, 6–10 March, Munich, Germany, 93–93.
162. Ventroux, N., Blanc, F. and Lavenier, D. A low complex scheduling algorithm for multiprocessor system-on-chip. *Proceeding of Parallel and Distributed Computing and Networks*, 15–17 February 2005, Innsbruck, Austria
163. Verdoolaege, S., Nikolov, H. and Stefanov, T. PN: A tool for improved derivation of process networks. *EURASIP Journal on Embedded Systems*, Volume 2007, Article ID 75947.

164. Vergnaud, T., Pautet, L. and Kordon, F. Using the AADL to describe distributed applications from middleware to software components. *Proceeding of Ada-Europe 2005*, York, UK, 20–24 June 2005, 67–78.
165. Kronos model checking tool, http://www-verimag.imag.fr/TEMPORISE/kronos/
166. Sangiovanni-Vincetelli, A. and Martin, G. Platform-based design and software design methodology for embedded systems. *IEEE Design and Test*, 18(6), 2001; 23–33.
167. Sangiovanni-Vincetelli, A. et al. Benefits and challenges for platform-based design. *Proceeding of DAC 2004*, USA
168. VSI Alliance. http://www.vsia.com/
169. Wallace, G. K. The JPEG still picture compression standard. *Communications of the ACM, Special Issue on Digital Multimedia Systems*, 34(4), April 1991, 30–44
170. VxWorks, http://windriver.com/vxworks
171. Wolf, W. High-Performance Embedded Computing. Morgan Kaufmann, San Francisco, 2006
172. h264 open source code, http://www.videolan.org/developers/x264.html
173. Xue, L., Ozturk, O., Li, F., Kandemir, M. and Kolcu, I. Dynamic partitioning of processing and memory resources in embedded MPSoC architectures. *Proceeding of DATE 2006*, 6–10 March 2006, Munich, Germany, 690–695.
174. Yoo, S. and Jerrara, A. A. Introduction to hardware abstraction layers for SoC. *Proceeding of DATE 2003*, 3–7 March 2003, Munich, Germany, 336–337.
175. Youssef, M. W., Yoo, S., Sasongko, A., Paviot, Y. and Jerraya, A. Debugging HW/SW interface for MPSoC: Video Encode System Design Case Study. *Proceeding of DAC 2004*, 7–11 June 2004, San Diego, USA, 908–913.
176. Zhao, Y., Liu, J. and Lee, E. A. A programming model for time-synchronized distributed real-time systems. *Proceeding of the 13th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'07)*, Bellevue, WA, USA, 2007.

Index