



[Previous Issues](#)



Monday  
April 26, 2004  
From: EDACafe

**FREE FPGA News Updates** [click here](#)  
 Technical Papers from Mentor, Xilinx, Altera, others  
 Events | Industry News



## About This Issue



## Behavioral Synthesis

April 19 - 23, 2004 **By Dr. Jack Horgan**

Read business product alliance news and analysis of weekly happenings

Last week's editorial was on Electronic-System Level (ESL) Design. Most vendors in this space have products built upon the SystemC language whose development is being managed by the Open SystemC Initiative (OSCI). The language and the organization are described later in this document.

This week's editorial is about Behavioral Synthesis currently being promoted by Forte Design System and Bluespec, Inc. While they disagree on the choice of language (SystemC and SystemVerilog respectively), they see considerable benefits of employing higher level of abstraction in terms of productivity and correct designs. They differ from ESL vendors in that they are targeting the hardware design engineer. They question whether one can make architectural evaluations and decisions like hardware/software partitioning without considering hardware details.

Bluespec point of view is that attempting to synthesize architecture from a high-level behavioral model is an intractable problem. No matter how good the software, it is no substitute for a hardware engineer's intuitions regarding the top level architecture and design choices. In Bluespec, all state elements are explicitly defined by the designer. "We automate hardware generation, not design choices."

Brett Cline, VP of Marketing for Forte Design, gave the rather trivial example of a function which accepts four inputs (a,b,c,d) and return the results of the calculation.

$$Y = (a*b)+c)*(d*e)$$

If one were to use two multipliers, one could schedule the operational sequence to perform the two internal multiplies (a\*b and d\*e) in parallel, followed by the addition and then the final multiplication. With only a single multiplier, one would schedule the first multiplication, then the addition and second multiplication in parallel, followed by the final multiplication. One sequence would be faster but with more silicon area than the other.

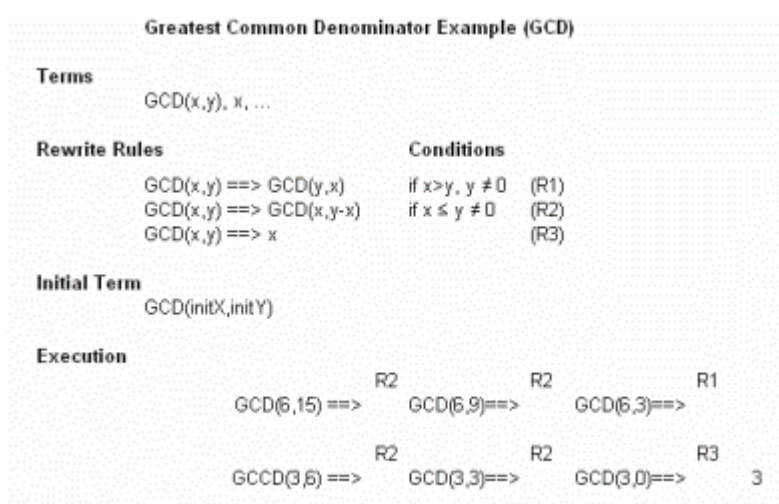
Bluespec discussed with me a more elaborate example of an IP Address Lookup where packets are routed based upon longest prefix match of the packets IP address. They identified three possible architectural alternatives: statically scheduled memory pipeline, straight pipeline with uncoordinated memory references and circular pipeline for 100% memory utilization. Each approach would have different area and speed characteristics. In any realistic design there would be a myriad of choices that would swamp any automated approach. They argue that the hardware designer must direct these choices.

Both vendors offer Behavior Synthesis that generates RTL code - the path to Silicon. Previous attempts in this space, even by major EDA companies, have failed giving the approach a poor image that these companies must overcome.

Bluespec Inc. was founded in June 2003 and soon raised \$4 million in VC funding. The company has 16 employees. I interviewed CEO Shiv Taskir, CTO Rishiyur Nikhil and Director of Marketing George Harper. Shiv had been SVP S&M at Viewlogic. The rest of the management team consists of EDA veterans. The technology is based upon patented work by IBM Professor Arvind, who is Bluespec cofounder and board member. Arvind had been CEO at Sandburst but is now back teaching at MIT after a two year sabbatical. The firm has not yet announced a product but there are a number of pilot studies. The marketing plans calls for focusing on Quality of Results by offering proof points for the technology. Expect upcoming announcements.

At the core of Bluespec's compiler technology is the introduction of the application of Term Rewriting Systems (TRS) to hardware description, the compilation of TRSs to high-quality RTL and formal verification through mathematical transformation ("correct by construction"). Term Rewriting Systems are a well-understood formalism from computer science. A TRS consists of "terms" which describe hardware states, and "rules" which describe behavior. A "rule" captures both a state-change (an "action") and the conditions under which it can occur. As the rules in a Term Rewriting System have atomic semantics, analysis of hardware can be done even though it may be highly concurrent and complex. TRS semantics and transformations enable the efficient mapping of the input into scheduled, optimized RTL, where control logic is synthesized for correct-by-compiler construction.

The example below explains the concept of TRS but not the actual syntax. A term represents a state element such as a flip-flop or register. Any legal behavior is explainable in terms of a sequence of atomic actions on the state. If there are N rules, a Scheduler analyzes all the rules, determines the ones that satisfy their conditions and fires them in parallel. The atomicity characteristic of the language supports modeling concurrency, thereby eliminating race conditions.



Bluespec employs the SystemVerilog language to create an industry standards-based design environment that significantly raises the level of abstraction for hardware design while retaining the ability to automatically synthesize high quality RTL, uncompromised in speed, power or area.

The Bluespec compiler accepts code based upon SystemVerilog and certain extensions. Some of these extensions have already been accepted by Accellera for inclusion in the standard. Bluespec will provide a style guide. The compiler will produce cycle-accurate C models and RTL. The RTL may be seamlessly fed into logic synthesis or into simulation packages from third parties. Existing RTL may be accepted by the compiler as called or calling code.

The company rejects the notion that firms should compromise on quality of results in order to gain productivity from use of higher levels of abstraction. They believe that the results of behavior synthesis must approach the results of hand coded RTL in order for hardware designers to adopt the methodology.

Bluespec believes that previous attempts at behavioral synthesis failed because their level of abstraction was too high and that the language used was not accepted within the standard making community. The earlier failures are seen as an obstacle for future acceptance.

### Forte Design Systems

The company was formed by the merger of CynApps Inc and Chronology Corporation in March 2001. Headquartered in San Jose, the firm has ~40 people. In April 2003 Forte raised \$9 million in series B funding on top of \$4 million initial funding.

Using industry-standard SystemC, algorithms expressed in C or C++ can be used as the basis for implementation. SystemC constructs are used to represent hardware-specific features, such as module hierarchy, bit-accurate data types, interface protocol, and reset behavior. Mike Meridith, vice president of technical marketing at Forte Design Systems, serves as executive director of the Open SystemC Initiative.

Forte's Cynthesizer automatically creates an RTL datapath, Finite State Machine (FSM), and control logic to implement the given algorithm. Information from the target technology library is used to ensure predictable timing closure. RTL can be generated in SystemC for simulation or in Verilog optimized for efficient logic synthesis.

High-level directives, such as latency, pipelining, and loop unrolling, are entered in a separate file so that implementations with differing characteristics may be generated without changing the source code of the original algorithm. This capability combines with the automation in the Behavioral Design Workbench to make it easy to create 5, 10, or more fully verified RTL implementations using different micro-architectures to achieve different latency and area characteristics. Cynthesizer also integrates with downstream logic synthesis tools. It produces RTL with a consistent style that is optimized for efficient logic synthesis

By designing at the highest possible level of abstraction throughout the design process, problems with the algorithm or implementation can be found much earlier. For example, it is much easier to isolate, debug, and fix a protocol error at the algorithmic level than it is at the register-transfer level.

### Stages of the behavioral synthesis process

Lexical processing parses the high-level language source code and transforms it into an internal representation

Algorithm optimization using techniques such as common subexpression elimination and constant folding

Control/Dataflow analysis yields a Control/Dataflow Graph (CDFG)

Library processing reads the available libraries and determines the functional, timing, and area characteristics of the available parts

Resource allocation establishes a set of functional units

Scheduling introduces parallelism and the concept of time. It transforms the algorithm into an FSM representation

Functional unit binding assigns the operations of the algorithm to specific instances of functional units from the library

Register binding

Output processing writes out the datapath and finite state machine resulting from all of the previous steps as RTL source code in the target language

Brett Cline gives two principal reasons why the use of higher level abstractions had not taken off before. First, there was no need, i.e. firms could produce designs without changing methodology—without moving up a level of abstraction. In last week's editorial on ESL I made the point of "No pain, no sale!" Today the challenge of meeting TTM pressures with increasing complex designs is generating a considerable amount of pain. Second, there was no path-to-implementation, no path-to-silicon from ESL. Brett believes the starting point for new designs is the algorithms, the functional descriptions, typically written in C or C++. Since Forte's products are based upon SystemC, they provide a natural progression to RTL. Brett sees Behavioral Synthesis potentially as a huge catalyst for ESL rather than as an alternative.

Brett also agrees with Bluespec that the results must be competitive with hand coded RTL. This will be the test that these two vendors must meet, if there are to be successful.

## Accellera

[Accellera](#) is a non-profit electronics industry organization formed in March 2000 from the unification of Open Verilog International (OVI) and VHDL International (VI). The organization exists to drive worldwide development and the use of standards required by systems, semiconductor and design tools companies, which enhance a language-based design automation process. This includes support of technical groups involved with developing standards for the IEEE 1364 Verilog HDL and IEEE 1076 VHDL, Verilog RTL subset, VHDL RTL subset, Design Constraints Description Language, IEEE 1481 Delay and Power Calculation System (DPCS), the Advanced Library Format (ALF), and the System Level Design Language (SLDL) Initiative. Accellera forwards completed standards to IEEE.

In December 2003 Accellera's Technical Committee formally accepted Bluespec's submission that addressed Tagged Unions and Pattern Matching for SystemVerilog 3.1a. The new elements raise the level of abstraction for design and improve the expressiveness and readability of the SystemVerilog language.

There has been a dispute between Accellera and IEEE when Accellera failed to meet an August IEEE deadline for the technology donation to be considered for inclusion the forthcoming IEEE 1364-2005 Verilog. If IEEE does not make some accommodation, the donations from Accellera would not become part of Verilog standard for several years. Donations from individual EDA vendors overlap those from Accellera. In April [Accellera Joined IEEE Standards Association \(IEEE-SA\)](#) to take part in and shape the direction of technology and its marketplace application. [Synopsys](#) and [Mentor Graphics](#) had also become Corporate Members earlier this month. Accellera's technical committee unanimously approved SystemVerilog3.1a standard timed for transfer to IEEE in June 2004.

In October 2003 [Synopsys Launched SystemVerilog Catalyst Program](#). The program is open to EDA vendors, silicon and verification IP companies, and training services providers to benefit mutual customers by advancing tool interoperability and the availability of IP using the Accellera SystemVerilog standard. More than 30 companies announced their support for SystemVerilog at the program's launch. Corporate members can gain early access to Synopsys' SystemVerilog-based tools, such as VCS and HDL Compiler, the front-end language compiler for Design Compiler, for development and support of their respective SystemVerilog tools, IP and training products. While not a member of the Catalyst program, [Cadence announced support for SystemVerilog](#).

## SystemVerilog

SystemVerilog, the recently ratified hardware description and verification language (HDVL) standard, is a major extension of the established IEEE 1364-2001 Verilog language, and was developed by Accellera to dramatically improve productivity in the design of large gate count, IP-based, bus-intensive chips. SystemVerilog is targeted primarily at the chip implementation and verification flow, with powerful links to the system level design flow. Its important SOC-targeted improvements over Verilog-2001 include:

- Support of modeling and verification at the "transaction" level of abstraction.
- SystemVerilog's Direct Programming Interface (DPI) enables it to "call" C/C++/SystemC functions, and vice versa.
- A set of extensions to address advanced design requirements
- A new mechanism to support assertion-based verification (ABV), enabling a "design for verification" methodology
- New features to support hardware models and testbenches that utilize object-oriented techniques

## SystemC

SystemC is the standard design and verification language built in C++ that spans from concept to implementation in hardware and software. SystemC, as a standard, enables and accelerates the exchange of system-level IP models and executable specifications using a common C-based modeling

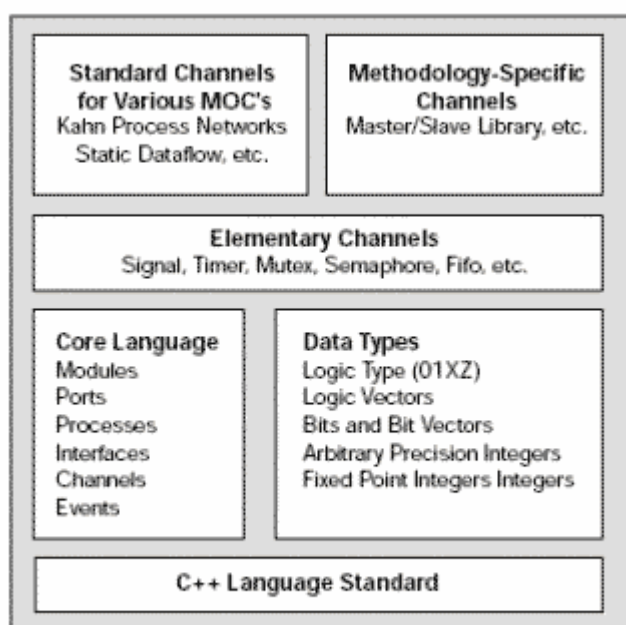
platform. Through the SystemC language and open source reference implementation, designers can create, validate and share models and executable specifications with other companies using standard ANSI C++, a language of choice for many architects, system designs and software developers. Most hardware designers are familiar with C.

The C++ general purpose object oriented programming language had to be extended in terms of hardware data types, concurrency, notion of time, and reactive behavior and communication.

SystemC 1.0 uses modeling constructs, e.g. modules, ports and signals, similar to those used for RTL and behavioral modeling. Structural design hierarchies can be built by instantiating modules within other modules. Communication of data between modules are enabled by ports and signals, All ports and signals are declared to have a specific data type such as single bits, bit vectors, characters, integers, vectors of integers and four-state logic signals

SystemC 1.0 models concurrent behaviors using processes. A process can be thought of as an independent thread of control which resumes execution when some set of events occur or some signals change, and then suspends execution after performing some action.

SystemC 2.0 provides a small but very general purpose modeling, termed the core language. This provides only a minimal set of modeling constructs for structural description, concurrency, communication, and synchronization. More specific models of computation, design libraries, modeling guidelines and design methodologies are added on top of this language foundation as required for system design. The core language consists of an event-driven simulator as the base. It works with events and processes.



SystemC 2.0 Architecture

SystemC 2.0 introduces a new set of features for generalized modeling of communication and synchronization. These are channels, interfaces and events. A channel is an object that serves as a container for communication and synchronization. Channels implement one or more interfaces. An interface specifies a set of access methods to be implemented within a channel, but the interface itself does not provide the implementation. An event is a flexible, low-level synchronization primitive that is used to construct other forms of synchronization.

Channels, interfaces and events enable designers to model the wide range of communication and synchronization found in system designs. Examples include HW signals, queues (FIFO, LIFO, message queues, etc.), semaphores, memories and busses (both as RTL and transaction-based models).

SystemC Verification (SCV) standard based on Cadence's TestBuilder C/C++ class library, will allow users to construct testbenches at a high level of abstraction. The major verification capabilities added on top of SystemC are constrained random testing, complex constraint solvers, data-structure creation and concurrency.

### The Open SystemC™ Initiative (OSCI)

The Open SystemC Initiative (OSCI) is an independent not-for-profit organization composed of a broad range of companies, universities and individuals dedicated to supporting and advancing SystemC as an open source standard for system-level design.

Firms may join OSCI as Corporate members (\$25K annual dues) who have some influence on strategic direction or as Associate Corporate Members (\$2K to \$10K depending on type and size) who can participate in technical working groups. The current corporate members are: ARM, Cadence, CoWare, Forte Design Systems, Fujitsu, Mentor, NEC and Synopsys.

The Open SystemC Initiative is based on an OSI-compliant Open Source licensing mode. The OSCI Steering Group and technical working groups ensure structured innovation by reviewing both shared modifications and community requirements and setting a technical roadmap. Working groups operate by building consensus regarding requirements, proposed new features, and implementation. The following technical working groups are currently active:

Language: Exploring improvements in the core SystemC language

Synthesis: Defining a synthesizable subset of the SystemC language

Transaction-level Modeling: Exploring infrastructure and methodology for the development of interoperable transaction-level models

Verification: Exploring infrastructure and methodology for using SystemC for functional verification

According to Michael Meredith, Executive Director, there are 20,500 registered licensees. Last year licensees were downloading from the web site ([www.systemc.org](http://www.systemc.org)) at the rate of 4,000/month and this year the rate is 5,000/month. The organization has had to upgrade their email system to accommodate the increased level of interest.

Michael is encouraged by the way groups of users are building their efforts on top of SystemC. He cited the case of the On-Chip Communication Network (OCCN) and Open Core Protocol International Partnership (OCP-IP). OCCN proposes an efficient, open-source research and development framework for the specification, modeling and simulation of on-chip communication architectures. OCP-IP is a ground-breaking industry association dedicated to making a common standard for IP core interfaces or sockets that facilitate "plug and play" SoC design.

SystemC's has backing from a number of major systems and semiconductor houses. Michael mentioned Sony and STMicroelectronics.

### Update on China

The US and China reached an accord on April 21 according to which China agreed to crackdown on counterfeiting and piracy of copyrighted material and intellectual property. China agreed to increase penalties including for online piracy, expand customs enforcement and to have government agencies stop using pirated software. China also agreed to indefinitely postpone plans to require that wireless communication devices comply with a Chinese encryption standard that was to go into effect on June 1. The Chinese government also promised to maintain "technology neutrality" with regard to 3G telecommunication standards. Remaining open issues between the two governments include currency policies, labor practices and taxes on semiconductor imports.

### Weekly Industry News Highlights

[Synopsys and ARM Team Up to Deliver System-Level Design Solution for ARM Powered Applications](#)

[NVIDIA Delivers Most Advanced Graphics Processor Ever, Designed Using Synopsys' Galaxy Design Platform](#)

[Four-Way AMD Opteron Processor-Based Servers From HP Deliver Exceptional Performance](#)

[Nassda's CRITIC Timing Analysis Tool Adopted by NVIDIA](#)

[Verisity Announces SpeXsim, First Combined Product with Axis Technology to Further Broaden Target Market](#)

[Verisity Provides Highest Value Verification Solution](#)

[Semiconductor Market to Outshine 2003 with 29% Growth in 2004 Reports In-Stat/MDR](#)

[Magma Announces Formal Verification Product Quartz Formal, Based on IBM Technology](#)

[More EDA in the News](#) and [More IP & SoC News](#)

### Upcoming Events...

--Contributing Editors can be reached by [clicking here](#).

You are registered as: [-EMAIL-].

CafeNews is a service for EDA professionals. EDACafe respects your online time and Internet privacy. To change your newsletter's details, including format and frequency, or to discontinue this service, please navigate to [http://www10.edacafe.com/nl/newsletter\\_subscribe.php](http://www10.edacafe.com/nl/newsletter_subscribe.php). If you have questions about EDACafe services, please send email to [edaadmin@ibsystems.com](mailto:edaadmin@ibsystems.com).

Copyright © 2004, Internet Business Systems, Inc. — 11208 Shelter Cove, Smithfield, VA 23420 — 888-44-WEB-44 — All rights reserved.