

Comprehensive SystemC

Standard Level - 5 Tage

Comprehensive SystemC ist ein 5-tägiges Training zur Einführung in SystemC™, eine C++ Klassenbibliothek zur System-Level-Modellierung. SystemC wird heute typischerweise verwendet, um Systemmodelle zu beschreiben, die sowohl Hardware- als auch Softwarekomponenten enthalten.

Das Kursprogramm umfasst die Grundelemente von SystemC und ihre Anwendung für das Transaction-Level Modelling. Der Kurs basiert auf Version 2.2 der SystemC Klassenbibliothek.

Der Kurs besteht aus 2 Modulen. Entwickler haben die Möglichkeit, die Module einzeln oder als kompletten 5-Tages-Kurs zu besuchen. Hardwaredesignern, die nicht auf umfassende Erfahrungen im Einsatz von C und C++ zurückgreifen können, wird die Teilnahme an beiden Modulen empfohlen.

- **Essential C++ for SystemC (1.-2. Tag)** vermittelt Entwicklern mit Grundkenntnissen in der Programmiersprache C in einer Schnelleinführung eine fundierte Grundlage für C++ sowie die erforderlichen Grundkenntnisse zum Erlernen von SystemC.
- **Fundamentals of SystemC (3.-5. Tag)** baut auf dieser Grundlage auf und bereitet die Entwickler auf die praktische Anwendung von SystemC für Transaction-Level Modelling vor. Der Kurs beschreibt die Klassenbibliothek v2.2 und ihre Anwendung für Systemmodellierung, Kommunikation, Hardware und Software auf dem Transaction-Level sowie Verfeinerungen in der Hardware-Software-Implementierung.

Die praktischen Übungen nehmen ca. 50% der Unterrichtszeit ein und basieren auf sorgfältig ausgearbeiteten Aufgaben, die die Kursteilnehmer in ihren neu erworbenen Kenntnissen herausfordern und das Erlernete untermauern.

Doulos ist weltweit führend in unabhängigem SystemC Know-How und ist seit dem Jahr 2000 aktiv auf dem Gebiet der auf SystemC basierenden Methodik. Wir haben SystemC Schulungen für Ingenieure aus über 100 Unternehmen weltweit durchgeführt. Dies beinhaltet auch direkten Kontakt zu Tool-Entwicklern in Unternehmen wie ARM, Cadence, CoWare, Mentor Graphics und Synopsys.

Zielgruppe

- Hardwaredesign-Ingenieure, die praktische Erfahrungen und Kenntnisse von SystemC für digitales Hardwaredesign und Simulation erwerben möchten
- Systemingenieure und -architekten, die Kenntnisse im praktischen Einsatz von SystemC für Systemmodellierung erwerben möchten
- Softwareentwickler, die bereits fundierte Kenntnisse in C/C++ besitzen, die jedoch Praxiserfahrungen mit den SystemC Klassenbibliotheken sammeln möchten

Kursinhalte

- Notwendige C++ Sprachmerkmale, um SystemC zu lernen und verstehen.
- Objektorientierte Programmier Techniken, wie sie in den SystemC Klassenbibliotheken verwendet werden.
- Die SystemC Grundelemente, Datentypen und Kanäle (channels)
- Wie man Systemmodelle am besten simuliert, debugged und validiert.

Continued...

Comprehensive SystemC

Standard Level - 5 Tage

- Wie man von RTL-Modellierung zur Transaction-Level-Modellierung gelangt
- Schreiben von Transaction-Level-Modellen für typische System-On-Chip-Strukturen
- Verfeinerung von SystemC-Modellen auf verschiedenen Abstraktionsebenen
- Überblick über die Hardwaresynthese mit SystemC (optional)
- Überblick über die SystemC Klassenbibliothek SCV (optional)

Voraussetzungen

Essential C++ for SystemC (Tag 1-2) Die Kursteilnehmer sollten über Grundkenntnisse in der Programmiersprache C verfügen (Funktionen, Variablen, Datentypen, Operatoren, Anweisungen). Dieses Modul eignet sich für Teilnehmer ohne Vorkenntnisse in C++, als Auffrischung für die, die wenig Vorkenntnisse in C++ haben, oder Hardwaredesigner mit Kenntnissen in VHDL oder Verilog®.

Fundamentals of SystemC (Tag 3-5) Praktische Erfahrungen mit C++ und objektorientierten Programmierkonzepten sind erforderlich und Grundkenntnisse des digitalen Hardwaredesigns werden empfohlen. Die vorherige Teilnahme am Doulos Essential C++ Kurs oder einem vergleichbaren Training ist erforderlich. Kursteilnehmer, die bereits Erfahrung in C++ haben, sollten Ihre Kenntnisse vor der Teilnahme mit den „**Voraussetzungen SystemC C++**“ vergleichen. Eine Liste mit Voraussetzungen ist bei Doulos erhältlich. Der Kurs eignet sich für Hardware-, Software- und Systemingenieure aus dem Bereich der Elektronikentwicklung. Um jedoch den maximalen Nutzen aus dem Kurs zu ziehen, sollten die Teilnehmer aktive Anwender einer Softwarehochsprache (Idealerweise C++) oder einer Hardwarebeschreibungssprache (VHDL oder Verilog®.) sein.

Wenden Sie sich bitte direkt an Doulos, um Ihre Erfahrungen und Kenntnisse in Hinblick auf diese Voraussetzungen zu erörtern und zu bewerten.

Kursunterlagen

Die Doulos Kursunterlagen sind für ihren umfassenden Informationsgehalt und die sehr benutzerfreundliche Präsentation allgemein bekannt. In ihrem Aufbau, Inhalt und ihrer Themenbehandlung sind sie einzigartig im EDA-Schulungsbereich, was sie zu begehrten Nachschlagewerken hat werden lassen. In den Kursgebühren sind enthalten:

- Kursskripte mit vollständigem Stichwortverzeichnis, die ein umfassendes Referenzhandbuch darstellen
- Ein Arbeitsbuch mit vielen praktischen Beispielen und Lösungen als Unterstützung bei der Wissensanwendung
- Doulos SystemC Golden Reference Guide für Sprache, Syntax, Semantik und Tipps

Comprehensive SystemC

Standard Level - 5 Tage

Struktur und Inhalt

Essential C++ for SystemC (2 Tage)

Tag 1

Learn about the differences between C and C++

From C to C++

The features added to C by C++ and the ANSI C-1999 standard • `const` • `bool` • Header files • Namespaces • The global and standard namespaces • Stream I/O

Functional and Pointers

Learn how functions and dynamic memory allocation have changed in C++ • Pass-by-reference • Function prototypes • Default arguments • Function overloading • Operator overloading • Static, automatic and dynamic storage • `new` • `delete`

The C++ Standard Library

Learn to make the most of the built-in standard classes • Container classes • Examples of using the standard vector class • Examples of using the standard string and stringstream classes

Classes and Objects

Learn the principles of object-based design • Information hiding • Abstract data types • Classes and objects • Public and private class members • Member functions • Scope resolution

Tag 2

Master the subtleties of object-oriented programming in C++

Class Members

Master the C++ mechanisms associated with classes • Constructors • Destructors • Copy constructors • Pointers versus objects • Friends • `this` • Overloading operators as members • Static members • Constant objects and members

Inheritance

Learn to exploit the power of object-oriented programming • Class relationships • Initializing sub-objects • The default constructor • Derived classes • Inheritance • Protected members • Up- and down-casting • Order of initialization

Virtual Functions

Delve deeper into object-oriented programming techniques • Overriding methods • Virtual functions • Polymorphism • Run-time type identification • Abstract base classes • Multiple inheritance

Further C++ Features

Advanced C++ features used in the SystemC class libraries • Function templates • Class templates • Implicit conversions • User-defined conversions • Exceptions

Comprehensive SystemC

Standard Level - 5 Tage

Fundamentals of SystemC (3 days)

Tag 3

Become proficient in using the features of SystemC

Introduction to SystemC

Learn the background to SystemC and how SystemC fits into the system-level design flow • The architecture of the SystemC release • The benefits and risks of adopting SystemC • The objectives of transaction-level modelling

Getting Started

Learn how SystemC source code is structured and how to organise files • SystemC header files and namespaces • Compiling and executing a SystemC model

Modules and Hierarchy

How to describe the structural connections between modules • Modules • Ports • Processes • Signals • Methods • Primitive channels • Module instantiation • Port binding

Processes and Time

Describing concurrency and the passage of time • SC_METHOD • SC_THREAD • Event finders • Static and dynamic sensitivity • Time • Events • Clocks • Dynamic processes

The Scheduler

Gain an insight into how SystemC manages the scheduling of processes and events • Starting and stopping simulation • Elaboration and simulation callbacks • The phases of simulation • Event notification • Event queues • `wait` and `next_trigger`

Tag 4

Learn to apply SystemC to modelling data, communication and busses.

Debugging and Tracing

Learn about the facilities provided by SystemC to ease debugging and diagnostics • Debugging techniques • The standard reporting mechanism • Error handling • Writing trace (`vcd`) files • Tracing buried signals and local variables • Using waveform display tools

SystemC Data Types

Data types for bit-accurate and hardware modelling • Signed and unsigned integers • Limited and finite precision integers • Assignment and truncation • Type conversion • Bit and part selects • Concatenation • Bit and logic vectors • Hexadecimal numbers • Avoiding common pitfalls • Bus resolution • Fixed point types

Interfaces and Channels

Learn how channels are used to abstract communication and create fast simulation models • Hierarchical, primitive and minimal channels • Interface method calls • SystemC interfaces • Port-less channel access • The SystemC object hierarchy • The class `sc_port` • Registering ports • How to make the most of ports, channels and interfaces

Comprehensive SystemC

Standard Level - 5 Tage

Tag 5

Exploration of the application of Transaction-Level Modelling

Bus Modelling

Learn the techniques required to write and use bus models in SystemC • Master and slave interfaces • The execution context of interface method calls • Blocking and non-blocking methods • Using events and dynamic sensitivity within channels • Multi-ports • Port binding policies • `sc_export`

Refinement

An example of refinement from a C algorithm through untimed and timed SystemC models down to a mixed hardware-software implementation • SystemC wrappers • Timing annotation • Using `sc_buffer` • Structural refinement, communication refinement, and data refinement

Adapters

Channel refinement using adapters • Events versus event finders • Instantiating and binding adapters

Transaction-Level Modelling

The transaction-level modelling space • The functional, architecture, programmers and verification views • TLM principles • Transactions versus transfers • Passing request and response objects • The OSCI TLM standard • The standard blocking, non-blocking and transport interfaces • Implementing the transport and protocol layers

Supplementary Subjects

Fixed Point Types

Fixed point word length and integer word length • Quantization modes • Overflow modes • Fixed point context • The type cast switch • Utility methods

Overview of SystemC Synthesis

RTL versus behavioural synthesis technology • The work of the OSCI synthesis working group • Synthesizable data types • Clocked threads and resets • Restrictions

Overview of the SystemC Verification Library

Introduction to and aims of SCV • Constrained random verification methodology • Extended data types to support introspection • Randomization • Transaction Recording

Verwandte Trainings

- Expert SystemC Modelling
- Expert SystemC Verification
- Modular SystemC (**nur Inhouse**)
- Comprehensive C++ (5 Tage)

Weitere Informationen

Um einen Platz zu reservieren oder Preisinformationen zu erhalten, wenden Sie sich bitte an das Doulos Sales Team.