

**Workshop-
& Seminar-
programm
2022**

**Ihr Partner für das Gesamtpaket
Training - Design Services - Produkte**

**PROGRAMMABLE LOGIC
COMPETENCE CENTER**

PLC2 GmbH

Hugstammweg 30
D -79112 Freiburg-Opfingen

Fon +49 (0) 7664-913 13-0
Fax +49 (0) 7664-913 13-99

E-Mail: info@plc2.de
Internet: www.plc2.com

Wissen kompetent vermittelt!

Unser Leistungsspektrum	Seite	5
Kontakt	Seite	6
Standorte	Seite	7
Ihre Vorteile	Seite	8
Unser Dienstleistungsangebot	Seite	10
Easy Start Reihe	Seite	13
Ausbildungsschwerpunkt FPGA	Seite	19
Ausbildungsschwerpunkt Embedded	Seite	52
Ausbildungsschwerpunkt DSP & Image Processing ...	Seite	97
Ausbildungsschwerpunkt Connectivity	Seite	102
Ausbildungsschwerpunkt Business	Seite	115
Seminare	Seite	120
Terminübersicht 2022	Seite	124
Allgemeine Geschäftsbedingungen	Seite	130

Anzeigen unserer Partner finden Sie auf den Seiten 12, 27, 34, 38, 42, 80/81, 92, 96, 101, 118 und der Umschlagrückseite

Lernen von Experten - erschließen Sie sich neue Perspektiven

Die wachsende Dynamik der digitalen Transformation und die daraus resultierenden Möglichkeiten zur Leistungsfähigkeit und -steigerung bestimmen weiterhin die künftigen Herausforderungen unserer Arbeitswelt und den Unternehmenserfolg. Um diesen Herausforderungen gerecht zu werden, sind immer leistungsfähigere Systeme notwendig, welche z.B. nicht nur durch Microcontroller abgedeckt werden können - auch FPGA-basierte Lösungen setzen sich immer weiter durch. Mit unseren Schulungen erwerben Sie die praxisrelevanten und theoretischen Kompetenzen, um diese Transformation meistern zu können.

Unser aktualisiertes und deutlich erweitertes Seminar-Programm bietet Ihnen einen breit gefächerten Überblick über unsere Schulungen zu unterschiedlichen Technologien. Erfahren Sie, wie Sie sich in einer zunehmend digitalisierten Welt nachhaltig erfolgreich positionieren können.

Wählen Sie zwischen Webinaren, Online- oder Inhouse-Schulungen oder unseren Präsenzveranstaltungen mit Workshop-Charakter. Alle Module sind einzeln buchbar und ein Einstieg jederzeit möglich. Selbstverständlich berücksichtigen und planen wir die Seminare auch ganz individuell auf Basis Ihrer konkreten Anforderungen oder wir bieten Ihnen eine Langzeitausbildung für Ihre persönliche Weiterbildung an.

Sie wünschen weitere Informationen? Sehr gerne beraten wir Sie bei der Wahl Ihres Seminars. Rufen Sie uns an unter Telefon +49 (0) 7664 91313-15. Ausführliche Informationen und Termine finden Sie rund um die Uhr auch auf unserer Website.

**Wir freuen uns auf Ihre Anmeldung.
Viele Grüße!**

Three handwritten signatures in black ink, arranged vertically. The top signature is the most stylized, the middle one is more fluid, and the bottom one is the most legible, appearing to read 'Stefan'.

Nikolai Krassin, Sina Krassin, Stefan Krassin

■ Wer sollte am Kurs teilnehmen?

Alle Entwickler die digitale Schaltungen entwickeln und verifizieren müssen. Wir bieten neben unseren XILINX spezifischen Workshops auch Kurse für Anwender an, in deren Unternehmen andere Technologien zum Einsatz kommen. Die Teilnahme an den Workshops ist der schnellste und effizienteste Weg die Funktionsweise und Entwicklung der FPGAs, SoCs/MPSoCs und CPLDs zu erlernen. Beispiele und Übungen ermöglichen es Ihnen Ihr Design schnell und effizient in einem Logik-Baustein, u.a. von XILINX, zu implementieren und zu verifizieren.

Die Workshops und Seminare werden nicht nur regelmäßig in verschiedenen Städten in Deutschland angeboten, sondern können auch vor Ort in Ihrer Firma gehalten werden. Nicht nur die Entwickler, sondern auch die Ingenieure und Verantwortlichen aus den Bereichen Fertigung und Qualitätssicherung sowie Produkt-Marketing können die Kurse besuchen, um die vielfältigen Einsatzbereiche der FPGA, SoC/ MPSoC und CPLD Produkte zu verstehen.

■ Was werden Sie lernen?

Sie werden nicht nur die neuesten XILINX Produkte kennen lernen, sondern auch die entsprechenden und optimalen Vorgehensweisen und Methoden bei der Entwicklung der Bausteine.

Sie werden lernen welche leistungsfähigen Eingabewerkzeuge für die Beschreibung einer speziellen Funktion zur Verfügung stehen und wie Ihre Schaltung implementiert und verifiziert werden kann. Sie werden lernen wie die Werkzeuge eingesetzt werden können, um Entwicklungsprozesse zu automatisieren oder auch wie diese Werkzeuge verwendet werden können, um die Implementierung zu optimieren.

■ Welche Voraussetzungen gibt es?

Je nach Workshop sind unterschiedliche Voraussetzungen erforderlich. Rufen Sie uns an, wir beraten Sie gerne.

Auf alle Fälle sollten Grundkenntnisse im Umgang mit dem PC vorhanden sein und der Teilnehmer sollte Erfahrung in der Entwicklung digitaler Schaltungen haben.

■ Mit welchen Schulungsunterlagen bzw. Hilfsmitteln werden Sie arbeiten?

Jedem Teilnehmer steht während der Schulung ein eigener PC/Laptop mit bereits vorinstallierter Designumgebung zur Verfügung. Zusätzlich stehen jedem Teilnehmer zur Bearbeitung der Übungen je nach Workshop verschiedene FPGA bzw. SoC/ MPSoC Evaluationboards zur Verfügung.

Quellenverzeichnis

© iStock.com/polesnoy (U1)

© patpics – stock.adobe.com (U2)

© agrus – stock.adobe.com (S. 7)

© gonin – stock.adobe.com (S. 38)

© Audrius Merfeldas – stock.adobe.com (S. 97)

© ZinetroN – stock.adobe.com (S. 114)

© Pasko Maksim – stock.adobe.com (S. 115)

© Markus Spiske – Unsplash (S. 118)

© savanno – stock.adobe.com (U3)

© Treecha – stock.adobe.com (U3)

PLC2 – Programmable Logic Competence Center GmbH

Standort Opfingen - Trainingszentrum

Hugstmattweg 30
79112 Freiburg-Opfingen
Fon: +49 76 64 9 13 13-0
Fax: +49 76 64 9 13 13-99
info@plc2.de

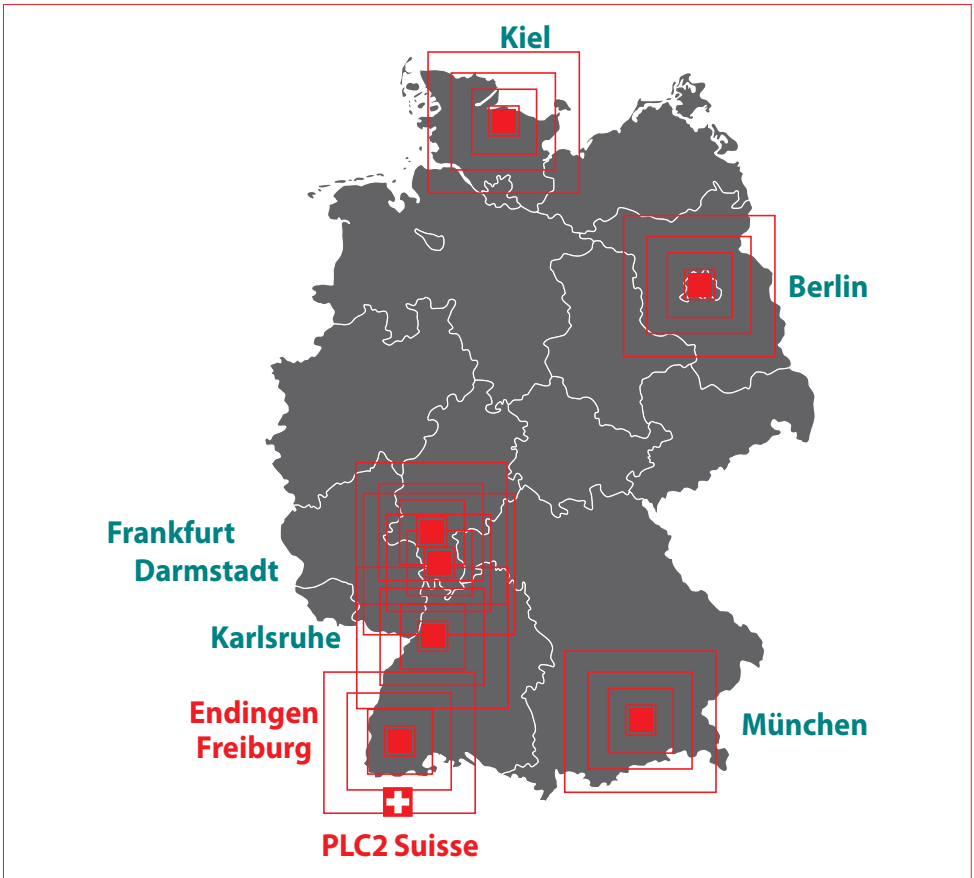


Standort Endingen - Entwicklungszentrum

Ersteiner Straße 19
79346 Endingen a. K.
Fon: +49 76 42 9 21 18-0
Fax: +49 76 42 9 21 18-29
info@plc2.de



Unsere Standorte in Deutschland und der Schweiz



Frankfurt

Im Vogelsgesang 5
60488 Frankfurt
Telefon: 0 69 74 30 76 96



PLC2 Suisse GmbH

Domherrenstraße 6
CH-4622 Egerkingen

Ihre Vorteile

Weniger Designiterationen durch VHDL

Wo Elektronik eine Rolle spielt, sind heute FPGAs fast immer unverzichtbar. Um auf dem Markt zu bestehen, muss man in kurzer Zeit neue FPGA-Architekturen entwickeln können. FPGA-Designer sind jedoch selten, denn der allgemeine Ingenieurmangel ist hier zusätzlich verschärft: An deutschen Hochschulen wird der Umgang mit FPGAs kaum oder nicht ausreichend gelehrt. Es gibt wenig Nachwuchs und so gut wie keine qualifizierten Bewerber auf dem Stellenmarkt. Sie erhalten eine umfassende Ausbildung und Zusatzqualifikation. Als FPGA-Experte erwarten Sie eine abwechslungsreiche und eigenständige Tätigkeit. Spezialisiertes Know-how sichert Ihren Arbeitsplatz.

Bei der Anwendung der am meisten geeigneten Designmethoden und Vorgehensweisen werden Sie die Zeit für die Implementierung und Verifizierung des Bausteins minimieren. Die Hardware-Beschreibungssprachen VHDL oder Verilog bieten hierfür die besten Voraussetzungen komplexe Schaltungen effizient zu beschreiben. Sollten doch Designänderungen notwendig sein, werden Sie lernen wie Sie diese Änderungen sicher und schnell in den Baustein implementieren können.

Schneller Marktzugang durch XILINX Technologie

Der schnelle Marktzugang ist wahrscheinlich einer der Hauptgründe warum Sie XILINX Technologie einsetzen wollen. Die Teilnahme an einem PLC2-Workshop wird es Ihnen ermöglichen die Produkte schnell auf den Markt zu bringen und vom schnellen Markteintritt zu profitieren. Beginnen Sie die Entwicklung des Bausteins während des Workshops. Bringen Sie Ihre aktuelle Entwicklung zur Veranstaltung mit und besprechen Sie die Aufgabenstel-

lungen mit dem Kursleiter. Die Zahl der Teilnehmer ist begrenzt, so dass diese Zeit zur Verfügung steht.

Verkürzte Einarbeitungszeit

Ausführliche Designbeispiele / Tutorials und Dokumentation der Hersteller helfen Ihnen Ihre Entwicklung früher abzuschließen. Allerdings ist der Besuch eines PLC2-Workshops mit kompetenter, konzentrierter und interaktiver Wissensvermittlung durch erfahrene Referenten weitaus effektiver, als das Lernen im Selbststudium. Anstatt durch die Alltagsarbeit stets unterbrochen zu werden, können Sie jetzt in kurzer Zeit zu einem Experten in der Entwicklung und im Einsatz der XILINX Technologien, sowie der Hardware-Beschreibungssprachen VHDL / Verilog / SystemVerilog werden.

Niedrigere Produktionskosten

Sie werden lernen, wie die Bausteine bestmöglich ausgenutzt werden können. Die Hardware-Beschreibungssprachen ermöglichen Ihnen mehr Logik in die Bausteine effizient unterzubringen und/oder höhere Taktraten zu erzielen. Somit können Sie die gesamten Produktionskosten Ihrer Entwicklung optimieren.

Qualität und Zuverlässigkeit

Bei der Verwendung effektiver Simulations- und Verifikations-Techniken können Sie die ordnungsgemäße Funktion des Bausteins einfach, sicher und schnell nachweisen. Die höhere Qualität Ihrer Entwicklungen kann die Änderungs-/Reparatur- und Pflege-Kosten entscheidend verringern.

Einsparung von Zeit und Geld

Die Teilnahme an PLC2-Workshops ist eine Investition, die sich bereits nach der ersten FPGA/CPLD bzw. SoC Entwicklung bezahlt macht. Wegen der kurzen zur Verfügung stehenden Zeitdauer, vermitteln die PLC2-Workshops das Wissen in einer

fokussierten und konzentrierten Form. Die durchgeführten Übungen stellen sicher, dass erlernte Vorgehensweisen praktisch angewendet werden und damit im Gedächtnis haften bleiben. Der Teilnehmer wird ein HDL- und XILINX-Experte und steht damit auch anderen Mitarbeitern im Hause zur Verfügung. Bei der Teilnahme an einer PLC2-Veranstaltung in Ihrer Nähe oder bei der Durchführung einer In-House Schulung bei Ihnen vor Ort können die Reisekosten gesenkt werden.

Übungen am Laptop/PC

Es werden bei allen Workshops praktische Übungen durchgeführt, um das Erlernte zu vertiefen. In der Regel steht jedem Teilnehmer ein Laptop bzw. PC während des Workshops zur Verfügung.

Referenten

Die Referenten haben langjährige Erfahrungen in Entwicklung und Ausbildung. Sie sind unter anderem von XILINX besonders geschult und ausgebildet worden, um ihr Wissen den Teilnehmern effizient und kompetent vermitteln zu können.

Schulungsunterlagen

Jeder Teilnehmer erhält einen kompletten Satz speziell ausgearbeiteter Schulungsunterlagen die auch später als Nachschlagewerke genutzt werden können. Die Schulungsunterlagen werden ständig überarbeitet und an Veränderungen angepasst und sind daher immer „up-to-date“.

Anmeldung

Sie haben die Möglichkeit sich direkt für einen Kurs online anzumelden. Bei weiteren Fragen können Sie uns gerne per Email oder Telefon kontaktieren.

Zertifikate

Als Bestätigung für die Teilnahme an einem unserer Workshops erhält jeder Teilnehmer ein PLC2-Zertifikat, das vom Referenten unterschrieben wird.

Informationen

Im Internet sind wir unter www.plc2.com zu finden. Dort erhalten Sie Auskunft über aktuelle Terminänderungen, sowie Neuerungen in unserem Workshop- und Seminarangebot.

Unser Dienstleistungsangebot

■ Veranstaltungsablauf

Beginn: 9.00 Uhr, Ende: 17.00 Uhr

■ Webinare

Unter www.plc2.com finden Sie kostenlose Webinare, live und on-demand, zu aktuellen Themen rund um die XILINX Technologien.

■ Seminare

Mit den XILINX Distributoren Avnet Silica und EBV Elektronik bieten wir kostenfreie Informations- und Update-Seminare an. Gerne beraten wir Sie auch bei der richtigen Wahl eines Seminars. Informationen dazu finden Sie in der Rubrik „Seminare“.

■ PLC2-Workshops

PLC2 GmbH ist ein autorisiertes XILINX Design- und Trainingszentrum, das auf die Entwicklung, Schulung und Beratung im Einsatzbereich der XILINX Technologien spezialisiert ist. Während unserer regelmäßig stattfindenden Workshops schulen wir die Anwender sowohl in der Designmethodik als auch in der Architektur der neuesten Bausteine bzw. in der physikalischen Implementierung und der Designverifikation. Im Rahmen dieser Workshops wurden bereits mehr als 7000 Entwickler im Einsatz der FPGA/CPLD und SoC Technologien geschult. Durch unsere enge und jahrelange Zusammenarbeit mit der Firma XILINX sind wir immer auf dem neuesten Stand der Technologie. Neben den speziell auf XILINX Technologien ausgerichteten Workshops haben wir auch Kurse im Programm, die für alle HDL Anwender konzipiert sind. In den entsprechenden PLC2-Workshops wie z.B. „Compact VHDL“, „Compact OSVVM“ oder auch „Compact Verilog“ bzw. „SystemVerilog“ werden Ihnen die Syntax und Befehle der Hardware-Beschreibungssprachen (HDL) vermittelt, unabhängig von der Technologie, die Sie einsetzen wollen.

■ PLC2-PowerWorkshops

Neben Veranstaltungen in Tagungshotels bietet PLC2 im Ausbildungszentrum Freiburg eine besondere Workshop-Reihe an. Unter der Bezeichnung „PowerWorkshops“ erhalten die Kursteilnehmer eine besonders praxisorientierte Ausbildung. Die PLC2-Power-Workshops dauern jeweils 5 Tage. Die Teilnehmer arbeiten mit PCs und Evaluation-Boards unter der Anleitung und Aufsicht erfahrener und kompetenter PLC2-Mitarbeiter. Dabei werden komplexe Aufgabenstellungen implementiert und die Funktionalität mit Hilfe der Simulation und des Evaluation-Boards nachgewiesen. Für die Unterbringung stehen beschauliche Hotels und Pensionen in unmittelbarer Nähe unseres Firmengebäudes zur Verfügung. Genießen Sie nach einem anstrengenden Workshoptag die „Badische Küche“ oder auch die Weine vom Tuniberg. Ein Streifzug in die nahe gelegene City von Freiburg ist ebenso empfehlenswert, wie die Landschaft um Freiburg herum.

■ In-House-Schulungen

Auch die Durchführung von In-House-Schulungen mit flexiblen und von bestehenden Workshops abweichenden Inhalten gehört zu unseren Schwerpunkten. Dabei können die Inhalte der Standardschulungen auf die besondere Situation oder auf den besonderen Bedarf hin angepasst werden. Neben der theoretischen Ausbildung können dabei die Anwender bereits während der In-House-Schulung projektspezifische Aufgabenstellungen bearbeiten und somit ihre Einarbeitungszeit entscheidend verkürzen. In-House-Schulungen sind Schulungen, die direkt vor Ort bei Ihnen in der Firma durchgeführt werden. Dies ist vor allem dann sinnvoll wenn Sie mehrere Entwickler haben, die zum gleichen Zeitpunkt oder aufgrund des selben Projektes geschult werden müssen. Die Inhalte können flexibel auf Sie abgestimmt werden, so erreichen Sie eine effektive Weiterbildung, die nur die Schwerpunkte beinhaltet, welche wirklich relevant sind. Nachfolgend

haben wir für Sie eine neue, nach Themenbereichen gegliederte Modulübersicht zusammengestellt, anhand derer Sie sich die In-House Schulung zusammenstellen können, die Sie benötigen. Neben den Schulungsunterlagen bringen wir alle zur Durchführung der Schulung benötigten Werkzeuge wie Laptops, Beamer sowie Evaluationboards bei Bedarf mit.

■ Online Training

Das gesamte PLC2 Schulungsprogramm steht Ihnen auch online zur Verfügung. Fragen Sie bei uns persönlich nach. Wir beraten Sie gerne..

■ Training-on-the-Job

Als eine besondere Form der In-House-Schulung können wir Ihnen darüber hinaus „Training-on-the-Job“ anbieten. Dabei unterstützen wir Sie bei Ihnen vor Ort bei der Entwicklung Ihrer Projekte. Neben der theoretischen Ausbildung durch einen unserer erfahrenen Referenten werden gleichzeitig aktuelle Aufgabenstellungen bearbeitet und Lösungswege erarbeitet. Durch diese Art des Trainings lernen Sie mit dem aktuellen Projekt und verkürzen gleichzeitig die Entwicklungszeit.

■ Coaching

Sie stehen mitten in Ihrer Entwicklung, doch dann tauchen unerwartete Probleme auf: die Software steigt aus, das Timingverhalten stimmt nicht, die

Simulation läuft nicht – es können noch weitere unerwartete Probleme auftreten. Buchen Sie einen unserer Mitarbeiter, er wird spezielle Lösungen ausarbeiten und Ihr Design voran bringen.

■ Designdienstleistungen

Neben den PLC2-Workshops und Seminaren bieten wir zusätzlich Designdienstleistungen an. Dabei ist die Schnittstelle zum Auftraggeber flexibel gestaltet, so dass ausgehend von einem Pflichtenheft oder einer Spezifikation, komplette Bausteine oder einzelne Submodule implementiert werden können. Gerne unterstützen wir Sie auch bei der Erstellung der Spezifikation. Profitieren Sie von unserer Flexibilität und Erfahrung in der Entwicklung komplexer Systeme.

■ XILINX Training Credits , TCs

Sie können die XILINX Kurse bei PLC2 auch mit sogenannten Training Credits (TCs), die Sie über ein XPA von XILINX bezogen haben, bezahlen. Die Abrechnung der TCs mit XILINX übernimmt PLC2, d.h. Sie haben keinerlei administrativen Aufwand. Sie melden sich lediglich bei PLC2 zu einer Schulung an, wir prüfen den Stand Ihrer noch vorhandenen TCs bei XILINX und bestätigen Ihnen Ihre Teilnahme entsprechend. Sollten Sie nicht mehr genügend TCs für einen Kurs haben, kann die Differenz selbstverständlich auch durch Zuzahlung beglichen werden.

Abgekündigte Bauelemente?

Sollten Sie Lieferschwierigkeiten mit abgekündigten Bauelementen haben so helfen wir Ihnen gerne weiter. Alles was wir dafür brauchen ist die Typenbezeichnung/Datenblatt des Bausteins. Unsere Spezialisten implementieren diese Funktion in einer dafür geeigneten CPLD- oder FPGA-Technologie. Selbstverständlich können noch andere von Ihnen gewünschte Funktionen zusätzlich mit in das CPLD/FPGA integriert werden.

Die PLC2 Easy Start Reihe

Mit der Easy Start Reihe bietet PLC2 GmbH schon seit Jahren ein besonderes Schulungskonzept an. Diese Schulungsreihe wurde entwickelt um gerade die Neulinge unter den FPGA bzw. SoC Anwendern gezielt beim Einsatz der FPGA bzw. SoC Technik unterstützen zu können. Dabei haben die Teilnehmer die Möglichkeit unter sachkundiger und professioneller Anleitung eigene FPGA/SoC Projekte zu realisieren und damit die ersten Schritte in Richtung selbständiger Entwicklung komplexer FPGAs/SoCs zu gehen.

Der Schwerpunkt dieser Reihe ist die praktische Arbeit mit modernen FPGAs/SoCs und weniger die Theorie. Dazu wird auf weitergehende Schulungen aus dem PLC2 Schulungsangebot verwiesen.

Der Workshop „Easy Start FPGA Vivado“ führt den Teilnehmer in die grundlegende Vorgehensweise bei der Entwicklung der FPGAs ein. Hier findet der Teilnehmer alle notwendigen Informationen rund um den Vivado Design Flow und kann


unmittelbar nach dem Workshop mit seiner ersten FPGA Entwicklung beginnen.

Die Workshops „Easy Start Embedded for ZYNQ UltraScale+ MPSoC Systems“ und „Easy Start Embedded for ZYNQ-7000 SoC Systems“ führen den Neuling in die Welt der Embedded Applikationen ein.

Die Entwicklung mit PetaLinux wird im Workshop „Easy Start Embedded PetaLinux“ behandelt und der „Easy Start Workshop“ vermittelt praxisnahe, direkt an der Hardware die notwendigen Entwicklungsschritte und weitere Details, um die neuen Xilinx KRIA SOM (System-On-Modules) erfolgreich in Projekten einzusetzen.

Die Anmeldegebühr zu einem der oben genannten Workshops aus der Easy Start Reihe beinhaltet i.R. ein geeignetes Evaluation Board.

Jeder Teilnehmer erhält in seinem „Easy Start“ Paket neben den Schulungsunterlagen auch die Dokumentation für alle im Workshop durchgeführten Übungen inklusive Musterlösungen.

Easy Start Reihe	Seite	Dauer (Tage)	Kosten
Easy Start FPGA Vivado	14	2	€ 1.700 o. 18 TCs + € 350
Easy Start Embed. for ZYNQ-7000 SoC Systems	15	2	€ 1.700 o. 18 TCs + € 350
Easy Start Embed. ZYNQ US+ MPSoC Systems	16	2	€ 1.500 o. 18 TCs
Easy Start Embedded PetaLinux	17	2	€ 1.700 o. 18 TCs + € 350
Easy Start Kria KV260 Vision AI Starter Kit 	18	2	€ 1.700 o. 18 TCs + € 350

Workshop Easy Start FPGA Vivado

Der Workshop Easy Start FPGA Vivado beleuchtet die grundlegende Vorgehensweise bei der Entwicklung der FPGAs. Hier findet der Teilnehmer alle notwendigen Informationen um mit einer FPGA Entwicklung zu beginnen. In diesem Workshop wird das Entwicklungstool Vivado verwendet. Nach einer kurzen Einführung in die Schaltungstechnik und den Entwicklungsvorgang der FPGAs, konzentriert sich der Workshop auf die Synthese / Simulation, sowie die Implementierung und die Inbetriebnahme der FPGAs. Der Schwerpunkt des Workshops liegt auf der Realisierung von FPGA Projekten. Für eine erfolgreiche FPGA Implementierung benötigt der Anwender ein grundlegendes Verständnis für die FPGA Schaltungstechnik, die Beschreibung und Synthese mit VHDL sowie die FPGA Verifikation. Auch muss ein grundlegendes Verständnis der FPGA Technologie vorhanden sein. Im Rahmen dieses Workshops werden verschiedene in der

Praxis anzutreffenden Aufgabenstellungen beispielhaft vorgestellt und gelöst. Die ordnungsgemäße Funktion der FPGA Entwicklungen wird sowohl mit Hilfe der Simulation als auch real mit dem Evaluation Board nachgewiesen. Eine noch tieferegreifende Wissensvermittlung finden Sie auch in dem PLC2 PowerWorkshop Professional FPGA.

Anwendbare Technologien

XILINX FPGAs

Voraussetzungen

keine

Dauer und Kosten

2 Tage, € 1.700,- netto pro Teilnehmer inklusive XILINX Evaluation Board mit Musterlösungen sowie Schulungsunterlagen, Pausengetränken und Mittagessen

Agenda

Einführung in den FPGA Entwicklungsvorgang

- Design Entry
- Simulation
- Synthese
- Implementierung
- Programmierung
- Übungen

FPGA Schaltungstechnik

- Kombinatorische Schaltungen
- Getaktete Schaltungen
- Asynchrone / synchrone Schaltungen
- Implementierung von Taktstrukturen
- Design Richtlinien
- Übungen

Schaltungssynthese mit VHDL

- Das VHDL Sprachkonzept
- Der VHDL Prozess
- Beschreibung kombinatorischer Schaltungen
- Beschreibung getakteter Schaltungen
- Übungen

Interne Taktstrukturen mit DCM / PLLs

- Taktnetze
- Clock Management
- MMCM
- MMCM Betriebsarten
- Übungen

Steuerungen implementiert mit State Machines

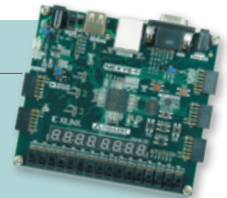
- Überblick FSMs
- Kodierung von State Machines
- Übungen

Schaltungssimulation mit VHDL

- Das VHDL Test Bench Konzept
- Das VHDL Timing Modell
- Erzeugung der Stimuli
- Übungen

Implementierung von internen Datenspeichern

- Deklaration von multidimensionalen Arrays
- Der Coding Style
- Realisierung der Speicher mit dem Core Generator
- Übungen





Workshop Easy Start Embedded for ZYNQ-7000 SoC Systems

Der Workshop „Easy Start Embedded for ZYNQ-7000 Soc Systems“ führt den FPGA Designer in die Welt des Embedded Designs im Zusammenhang mit der bewährten ZYNQ-7000 SoC Technologie. Hier lernt der Teilnehmer wie komplexe Embedded Anwendungen für den ZYNQ-7000 SoC Baustein realisiert werden können. Nach einer kurzen Einführung in die Architektur des ZYNQ-7000 SoC Bausteins beginnt der Teilnehmer mit der Implementierung seiner Embedded Anwendung. Der Schwerpunkt des Workshops liegt auf der Projektierung von Embedded Projekten mit der ZYNQ-7000 SoC Plattform und der Vitis Unified Development Plattform. Jeder Teilnehmer wird in den zwei Tagen alle wichtigen Schritte der Entwicklung mit Hilfe eines Evaluierungsboards durchführen: Hardware Plattform Generierung mit Wizard- Unterstützung, inkrementelle Hardware Ergänzungen sowie Software Library Generierung und Einbindung. Der Anwender lernt auch C-Applikationsprojekte unter Vitis zu erstellen und zu debuggen. Es werden verschiedene Hardware Plattformen erstellt und mit verschiedenen

Software Applikationen in Betrieb genommen.

Eine noch tieferegreifende Wissensvermittlung finden Sie auch im PLC2 PowerWorkshop „Professional ZYNQ-7000 SoC“, sowie in den 3-Tages Varianten „Compact ZYNQ-7000 SoC for Hardware Designers“ und „Compact ZYNQ-7000 SoC for Software Designers“ als alternative Empfehlung.

Anwendbare Technologien

XILINX ZYNQ-7000 SoCs

Voraussetzungen

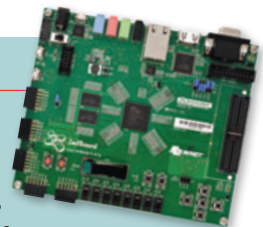
Grundlagenkenntnisse von Embedded Controllern
Grundlagenkenntnisse FPGA Technologie
Grundlagenkenntnisse Programmiersprachen VHDL und C

Dauer und Kosten

2 Tage, € 1.700,- netto pro Teilnehmer inklusive XILINX Evaluation Board mit Musterlösungen sowie Schulungsunterlagen, Pausengetränken und Mittagessen

Agenda

- ZYNQ-7000 SoC Architecture Basics
- IP Integrator and the PS Configuration Wizard
- **Lab 1:** Hardware Construction Using the IP Integrator Tool
- Software Development Using Vitis
- **Lab 2:** Adding and Downloading Software
- Adding Hardware to an Embedded System
- **Lab 3:** Adding IP to a Hardware
- Designing a Custom AXI Peripheral
- **Lab 4:** Building Custom AXI IP for an Embedded System
- Adding Custom IP to the Embedded System
- **Lab 5:** Integrating a Custom Peripheral
- Interrupts
- **Lab 6:** Software Interrupts
- Application Debugging
- **Lab 7:** Debugging





Workshop Easy Start Embedded for ZYNQ UltraScale+ MPSoC Systems

Der Workshop „Easy Start Embedded for ZYNQ US+ MPSoC“ führt den FPGA Designer in die Welt des Embedded Designs im Zusammenhang mit der bewährten ZYNQ UltraScale+ MPSoC Technologie. Hier lernt der Teilnehmer wie komplexe Embedded Anwendungen für den ZU+ MPSoC Baustein realisiert werden können. Nach einer kurzen Einführung in die Architektur beginnt der Teilnehmer mit der Implementierung seiner Embedded Anwendung. Der Schwerpunkt des Workshops liegt auf der Projektierung von Embedded Projekten und der dazugehörigen Vitis Unified Development Platform. Jeder Teilnehmer wird in den zwei Tagen alle wichtigen Schritte der Entwicklung mit Hilfe eines Evaluierungsboards durchführen: Hardware Plattform Generierung mit Wizzard- Unterstützung, inkrementelle Hardware Ergänzungen sowie Software Library Generierung und Einbindung.

Der Anwender lernt auch C-Applikationsprojekte in der Vitis Plattform IDE zu erstellen und zu debuggen. Es werden verschiedene Hardware Varianten erstellt und mit ver-

schiedenen Software Applikationen in Betrieb genommen. Eine noch tieferegreifende Wissensvermittlung finden Sie auch im PLC2 PowerWorkshop „Professional ZYNQ UltraScale+ MPSoC“, sowie in den 3-Tages-Varianten „Compact ZYNQ UltraScale+ MPSoC for Hardware Designers“ und „Compact ZYNQ UltraScale+ MPSoC for Software Designers“ als alternative Empfehlung.

Anwendbare Technologien

XILINX ZYNQ UltraScale+ MPSoC

Voraussetzungen

Grundlagenkenntnisse von Embedded Controllern

Grundlagenkenntnisse FPGA Technologie

Grundlagenkenntnisse Programmiersprachen VHDL und C

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

ZYNQ UltraScale+ MPSoC Architecture Basics

- Lab 1: Exploring the Architecture of the ZYNQ UltraScale+ MPSoC

Overview of Embedded Hardware Development

- Lab 2: Driving the IP Integrator Tool

Overview of Embedded Software Development

- Lab 3: Driving the Vitis Toolchain

Designing a Custom AXI Peripheral

- Lab 4: Building a Custom AXI IP

Standalone Software Platform Development

- Lab 5: Application Development

Interrupts – Hardware and Software Support

- Lab 6: ZU+ MPSoC Interrupts

Software Debugging with Vitis

- Lab 7: Application Debugging

Zynq UltraScale+ MPSoC Boot and Configuration

- Lab 8: ZU+ MPSoC Boot and Configuration

Workshop

Easy Start Embedded PetaLinux

XILINX bietet mit PetaLinux eine Open Source Lösung für ZYNQ-7000 Cortex-A9 wie auch MicroBlaze Prozessoren. Diese XILINX-Lösung wird ergänzt durch kommerziellen Support, Dokumentation, Wartung und Pflege, wo im Open-Source Kontext große Lücken klaffen.

Dieser Workshop vermittelt eine kurze Einführung in die Grundlagen eines Embedded Linux Betriebssystems, PetaLinux bietet eine einfachere Form der Kernel Konfiguration basierend auf Shell Skripte, einfache Methoden, QEMU Simulation, Firmware Packing, Templates sowie die Dokumentation.

Die Pflege von Open-Source basierenden Linuxsystemen in der XILINX Toolchain ist mit PetaLinux von großem Vorteil.

Auch wird es Linux Einsteigern leicht macht, den Kernel zu konfigurieren, die Hardwareperipherie zu unterstützen und das Embedded System schnell und effizient mit User-Applikationen zu entwickeln.

Alle wesentliche Schritte von der Kernel Konfiguration bis zum Einbinden von einer PL-Hardware-Peripherie mit Treiberunterstützung werden durchgeführt. Das in den Übungen verwendete ZYNQ Evaluierungsboard nehmen die Teilnehmer mit nach Hause.

Anwendbare Technologien

XILINX FPGAs und ZYNQ Derivate

Voraussetzungen

Grundkenntnisse Embedded Systems Software Design,

Grundkenntnisse der Programmiersprache C

Grundkenntnisse von Linux sind von Vorteil

Dauer und Kosten

2 Tage, € 1.700,- netto pro Teilnehmer inklusive ZYNQ-Evaluierungsboard mit Musterlösungen sowie Schulungsunterlagen, Pausengetränken und Mittagessen

Agenda

Embedded Linux Overview

- Lab: A First Look

PetaLinux for MicroBlaze and Cortex-A9 in ZYNQ

- Lab: Build and Boot an Image

Introduction to the PetaLinux Tools

Application Development and Debugging

- Lab: Application Development and Debug

Networking and TCP/IP

- Lab: Networking and TCP/IP

Device Drivers, User Space I/O, and Loadable Modules

- Lab: Accessing Hardware Devices from User Space

Custom Hardware Development and Interfacing

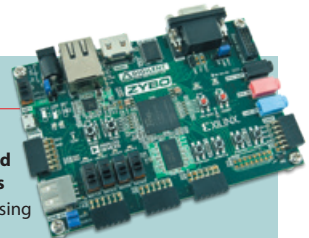
- Lab: Processor Configuration and Board Bring Up

Board Bring Up with the Vivado Design Suite and PetaLinux Tools

- Lab: Custom Hardware Development

Device Drivers for the Hardware

- Lab: Custom Driver Development



Workshop

Easy Start with the Kria KV260 Vision AI Starter Kit

NEU

Dieser Kurs ist Ihr Einstieg in das Arbeiten mit dem Xilinx Kria™ System-on-Module (SOM) und Kria KV260 Vision AI Starter Kit. Das KV260 Starter Kit ermöglicht Ihnen direkt nach dem Auspacken ohne FPGA-Kenntnisse die Beschleunigung von Vision-basierten Anwendungen. Sie erhalten einen Überblick über den Workflow mit den Tools Xilinx Vitis und PetaLinux.

Der Kurs bietet auch Informationen, wie Sie Ihre eigenen Hardware- und Softwarekomponenten von Grund auf neu erstellen können, um ein KI-Modell anpassen und eine Benchmark Analyse durchführen. Darüber hinaus werden auch Guidelines für Anwendungen von Eigenentwicklung behandelt.

Anwendbare Technologien:

Xilinx Kria™ System-on-Module (SOM) und Kria KV260 Vision AI Starter Kit

Voraussetzungen

Grundkenntnisse in der Programmiersprache C/C++

Dauer und Kosten

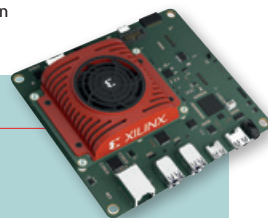
2 Tage, € 1.700,- netto pro Teilnehmer inklusive Xilinx Kria KV260 Vision AIE Starter Kit mit Musterlösungen, ausführlicher Schulungsunterlagen sowie Pausengetränke und Mittagessen

Agenda

- Introduction and Agenda
- Xilinx Kria System-on-Module (SOM) Overview
- Xilinx Zynq MPSoC Architecture
- Xilinx Kria KV260 Vision AI Starter Kit Overview
- Getting Started with the Vision AI Starter Kit
- Introduction to Vitis Video Analytics SDK (VVAS)
- Accelerating Applications with the KV260 Vision AI Starter Kit
- Building the Hardware and Software Design Components
- Xilinx PetaLinux Flow
- Customizing the AI Models
- Kria SOM Carrier Card Design Guide
- Xilinx Vitis Acceleration Libraries

Labs:

- Running the Demo Application with the Kria KV260 Starter Kit
- Installing the Xilinx Software Tools
- Accelerating Applications with the KV260 Vision AI Starter Kit
- Building the Hardware and Software Design Components



Ausbildungsschwerpunkt „FPGA“

Die in diesem Bereich zusammengefassten Workshops bilden das Fundament das Sie benötigen um mit dem FPGA Design beginnen zu können bzw. um die FPGA Designs zu optimieren. Das Angebot umfasst sowohl die Architektur der FPGAs, den Software Design Flow, HDL Synthese und HDL Verifikation sowie Design Optimierung. Manche der in diesem Abschnitt zusammengefassten Schulungen sind für FPGA Neueinsteiger, andere wiederum für erfahrene Entwickler geeignet.

Bewährte Schulungen im Bereich „FPGA“ sind:

- Designing with the XILINX -7 Family
- Designing with the XILINX UltraScale / UltraScale+ Family
- Continuous Integration
- Git
- Compact FPGA Schaltungstechnik
- Professional FPGA Schaltungstechnik
- Vivado Design Suite Static Timing Analysis and XILINX Design Constraints

Sie können sich jederzeit von unseren erfahrenen Trainern und Referenten im Vorfeld der Schulungs-Auswahl beraten lassen.



Ausbildungsschwerpunkt „FPGA“

FPGA	Seite	Dauer (Tage)	Kosten
Compact FPGA 7 Series	21	2	€ 1.500 o. 18 TCs
Designing with the Spartan-7 Family	22	2	€ 1.600 o. 18 TCs
Professional FPGA	23	5	€ 3.100 o. 45 TCs
Compact UltraScale/UltraScale+	24	2	€ 1.500 o. 18 TCs
FPGA Power Optimization	25	2	€ 1.500 o. 18 TCs
Dynamic Function eXchange (DFX)	26	2	€ 1.500 o. 18 TCs
Compact FPGA Schaltungstechnik	28	3	€ 2.100 o. 27 TCs
Professional FPGA Schaltungstechnik	29	5	€ 3.100 o. 45 TCs
Compact VHDL for Synthesis	30	3	€ 2.100 o. 27 TCs
Compact VHDL for Simulation	31	2	€ 1.500 o. 18 TCs
Professional VHDL	32	5	€ 3.100 o. 45 TCs
Compact VHDL Testb. & Verific. with OSVVM	33	3	€ 2.100 o. 27 TCs
Prof. VHDL Testbenches & Verif. with OSVVM	35	5	€ 3.100 o. 45 TCs
Compact Verilog	36	3	€ 2.100 o. 27 TCs
SystemVerilog – Adv. Verif. for FPGA Design	37	3	€ 2.100 o. 27 TCs
UVM Made Easy for FPGA Designers	39	2	€ 1.500 o. 18 TCs
Developing for Mission Critical FPGA & SoC	40	3	€ 2.100 o. 27 TCs
Git	41	3	€ 2.100 o. 27 TCs
Continuous Integration	43	5	€ 3.100 o. 45 TCs
Vivado Design Suite Tool Flow	44	1	€ 750 o. 9 TCs
Vivado Static Timing Analysis & Design Constr.	45	3	€ 2.100 o. 27 TCs
Professional VIVADO	46	5	€ 3.100 o. 45 TCs
Debugging Techniques Vivado Logic Analyzer	47	2	€ 1.500 o. 18 TCs
Tcl Schnellstart	48	1 o. 2	€ 750 o. € 1.500 o. 18 TCs
Advanced Vivado-Tcl-Scripting	49	3	€ 2.100 o. 27 TCs
Designing with the XILINX AMS Solution	50	2	€ 1.500 o. 18 TCs
SysML-Einführung für Systemingenieure	51	1	€ 750 o. 9 TCs

Workshop Compact FPGA 7 Series

Der PLC2-Workshop „Compact FPGA 7 Series“ schult den Neu- bzw. Wiedereinsteiger in der Funktionsweise und bestmöglichen Nutzung der Designressourcen der 7er-Serie der XILINX FPGAs.

Der Schwerpunkt dieses Workshops liegt auf der Beschreibung der grundlegenden Architekturelemente der aktuellen FPGAs. Nach einem Überblick wird im Detail auf den Aufbau der einzelnen Funktionselemente eingegangen wie z.B. die Configurable Logic Blocks (CLB), I/O-Blocks (IOB) für source- sowie systemsynchrone Übertragung im Single- oder Double-Data-Rate-Modus, DSP etc. Natürlich spielt dabei auf Grund der besonderen Bedeutung insbesondere die Taktstruktur eine große Rolle. Ferner werden dedizierte Hardwareblöcke (z.B. GTP und PCIe) kurz angesprochen. Außerdem werden geeignete Kodieretechniken diskutiert, welche der Synthese eine optimale Abbildung auf die Zielhardware erlauben. Die

grundlegenden Befehle der Beschreibungssprache VHDL sowie die allgemeine Vorgehensweise der Implementierung von FPGA-Designs sind nicht Bestandteil dieses Workshops. Hier wird auf die PLC2-Workshops „Compact VHDL“ bzw. „Professional VHDL“ verwiesen. Die theoretischen Inhalte werden durch praktische Übungen am PC abgerundet.

Anwendbare Technologien

7-Series FPGA-Technologies

Voraussetzungen

Grundkenntnisse in VHDL und Digitaltechnik

Grundkenntnisse im XILINX Design Tool Flow

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

CLB Architecture

- CLB Structure and Routing
- Slice Resources
- Distributed RAM/SRL
- Using Slice Resources

Slice Flip-Flops

- Control Sets
- Designing Resets
- Other Reset Considerations

Memory Resources

- Block RAM Capabilities
- FIFO Capabilities
- Using Block RAM Resources

DSP Resources

- 7 Series FPGA DSP Slice
- Pre-Adder and Dynamic Pipeline Control Advantages
- IP Support and Inference

I/O Resources

- SelectIO Electrical Resources
- SelectIO Logical Resources
- Power Savings
- Using SelectIO Resources

Clocking Resources

- Clock Networks and Buffers
- Clock Management Tile
- Usage Models
- Using Clock Resources

Memory Controller

- Phaser and I/O FIFOs
- Memory Controller
- Memory Interface Generator (MIG)

Dedicated Hardware

- Serial Gigabit Transceivers
- PCI Express Technology Interface
- XADC

HDL Coding Techniques

- Hierarchy
- Control Sets
- Synthesis Options

Workshop Designing with the Spartan-7 Family

Die XILINX FPGA Spartan Familie bietet das günstigste Preis-/Leistungsverhältnis der XILINX FPGA Technologie und wird daher insbesondere auch in Produkten von hoher Stückzahl eingesetzt.

Die Spartan-7 Technologie bietet eine Performancesteigerung von ca. 30% und Leistungsreduktion bis zu 50% im Vergleich zur vorherigen Spartan-6 Technologie.

Der Workshop führt Sie durch die Architektur der Technologie, um die Programmierung ob Ressourcen-schonend oder für hohe Performance optimieren zu können.

Im Workshop werden die Übungen am Rechner sowohl simulativ als auch mit Hardware-Evaluierungsboards durchgeführt.

Der FPGA Design Flow mit der Vivado Tool Suite, wie Projekt Management, HDL-Kodierung und best methods werden behandelt.

Insbesondere auch die Synthese- und Implementierungs-Optimierungen werden aufgezeigt und wie erstellte Kompilate zu analysieren sind.

Ein Spartan-7 Evaluierungsboard ist im Preis mit einbezogen.

Anwendbare Technologien

Spartan-7 FPGAs

Voraussetzungen

Grundkenntnisse in VHDL

Grundkenntnisse der Digitaltechnik

Dauer und Kosten

2 Tage, € 1.600,- netto pro Teilnehmer, Spartan-7

Evaluierungsboard, inklusive ausführlichen

Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

FPGAs Overview

- 7 Series Families
- Spartan-7 FPGAs
- FPGA Configuration

CLB Architecture

- CLB and Slice Resources
- Slice Flip-Flops
- Designing Resets

Memory Resources

- Distributed RAM/SRL
- Block RAMs und FIFO
- Using RAM Resources

DSP Resources

- Spartan-7 DSP Slice
- Arithmetic Support
- IP Support and Inference

I/O Resources

- SelectIO Interface
- SelectIO Interface Resources
- XADC and AMS

Clocking Resources

- Clock Networks and Buffers
- Clock Management Tile and PLLs
- Clock Usage Models

UltraFast Design Methodology

- Pin Planning
- HDL Coding Techniques
- Reset Management

Introduction to Vivado Design

- Vivado Design Flows
- Vivado Design Suite Project Mode
- Synthesis and Implementation
- Vivado IP Management
- FPGA Constraining

PowerWorkshop Professional FPGA

Der PLC2-PowerWorkshop „Professional FPGA“ schult den Neu- bzw. Wiedereinsteiger in der Funktionsweise und bestmöglichen Nutzung der Designressourcen der XILINX FPGAs.

Der Schwerpunkt dieses Workshops liegt auf der Beschreibung der grundlegenden Architekturelemente der aktuellen FPGAs. Nach einem Überblick wird im Detail auf den Aufbau der einzelnen Funktionselemente eingegangen wie z.B. die Configurable Logic Blocks (CLB), I/O-Blocks (IOB) für source- sowie systemsynchrone Übertragung im Single- oder Double-Data-Rate-Modus, DSP etc. Natürlich spielt dabei auf Grund der besonderen Bedeutung insbesondere die Taktstruktur eine große Rolle. Ferner werden dedizierte Hardwareblöcke (z.B. Gigabit Transceiver und PCIe) kurz angesprochen. Außerdem werden geeignete Kodieretechniken diskutiert, welche der Synthese eine optimale Abbildung auf die Zielhardware erlauben. Die grundlegenden

Befehle der Beschreibungssprache VHDL sowie die allgemeine Vorgehensweise der Implementierung von FPGA-Designs sind nicht Bestandteil dieses Workshops. Hier wird auf die PLC2-Workshops „Compact VHDL“ bzw. „Professional VHDL“ verwiesen. Eine Vorstellung und Diskussion der Timing Constraints rundet den Inhalt ab.

Anwendbare Technologien

UltraScale/UltraScale+ (ältere FPGA Familien auf Anfrage möglich)

Voraussetzungen

Grundkenntnisse in VHDL und Digitaltechnik
Grundkenntnisse im XILINX Design Tool Flow

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

CLB Architecture

- CLB Structure and Routing
- Slice Resources
- Distributed RAM/SRL
- Using Slice Resources

Slice Flip-Flops

- Control Sets
- Designing Resets
- Other Reset Considerations

Memory Resources

- Block RAM Capabilities
- FIFO Capabilities
- Using Block RAM Resources

DSP Resources

- DSP Slice
- Pre-Adder and Dynamic Pipeline Control Advantages
- IP Support and Inference

I/O Resources

- SelectIO Electrical Resources
- SelectIO Logical Resources
- Power Savings
- Using SelectIO Resources

Clocking Resources

- Clock Networks and Buffers
- Clock Management Tile
- Usage Models
- Using Clock Resources

Memory Controller

- Phaser and I/O FIFOs
- Memory Controller
- Memory Interface Generator (MIG)

Dedicated Hardware

- Serial Gigabit Transceivers
- PCI Express Technology Interface
- System Monitor

HDL Coding Techniques

- Hierarchy
- Control Sets
- Synthesis Options

Timing Constraints

- Accessing the Design Database
- Static Timing Analysis and Clocks
- Inputs and Outputs
- Timing Exceptions

Workshop Compact UltraScale/UltraScale+

Der PLC2-Workshop „Compact UltraScale“ schult den Neu- bzw. Wiedereinsteiger in der Funktionsweise und bestmöglichen Nutzung der Designressourcen der XILINX UltraScale FPGAs. Der Schwerpunkt dieses Workshops liegt auf der Beschreibung der grundlegenden Architekturelemente der UltraScale FPGAs. Nach einem Überblick wird im Detail auf den Aufbau der einzelnen Funktionselemente eingegangen wie z.B. die Configurable Logic Blocks (CLB), I/O-Blocks (IOB), DSP etc. Es wird zusätzlich auf die Design-Migration eines vorhandenen 7-Series Designs eingegangen. Natürlich spielt dabei auf Grund der besonderen Bedeutung insbesondere die Taktstruktur eine große Rolle. Ferner werden dedizierte Hardwareblöcke (z.B. GTX, GTY und PCIe) kurz angesprochen. Außerdem werden geeignete Kodieretechniken diskutiert, welche der Synthese eine optimale Abbildung auf die Zielhardware erlauben. Die grundlegenden Befeh-

le der Beschreibungssprache VHDL sowie die allgemeine Vorgehensweise der Implementierung von FPGA-Designs sind nicht Bestandteil dieses Workshops. Hier wird auf die PLC2-Workshops „Compact VHDL“ bzw. „Professional VHDL“ verwiesen. Die theoretischen Inhalte werden durch praktische Übungen am PC abgerundet.

Anwendbare Technologien

UltraScale/UltraScale+ Technologien

Voraussetzungen

Grundkenntnisse in VHDL und Digitaltechnik

Grundkenntnisse im XILINX Vivado Tool Flow

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Design Migration Software Recommendations

- Objectives

CLB Architecture and HDL Coding Styles

- Configurable Logic Block
- HDL Coding Techniques

Clocking Resources

- Resource Layout and Clocking Structure
- Clock Routing
- Clock Buffers
- Clock Management
- Using the Clock Resources

Memory and DSP Resources

- Block RAM Capabilities
- FIFO Capabilities
- DSP Capabilities
- Using RAM, FIFO, and DSP Capabilities

I/O Resources

- UltraScale Architecture Challenges and Solutions
- I/O Electrical and Physical
- I/O Logic and Clocking
- SelectIO Bank and Pin Names

FPGA Design Migration

- Objectives

Design Migration Case Study

- Migration Methodology
- XAUI Design Introduction
- Migrating the XAUI Design Case Study

Transceiver Overview

- Architecture Improvements
- Transceiver Wizard
- Transceiver Usage

Workshop FPGA Power Optimization

Eine Teilnahme an dem Workshop FPGA Power Optimization ermöglicht Ihnen, Ihre FPGA-Anwendungen besser zu optimieren, Sie könnten ein kleineres, günstigeres FPGA einsetzen, den Stromverbrauch reduzieren, oder ein günstigeres Kühlkonzept wählen, oder Systeme robuster gestalten. Das Beherrschen der Parameter wie der Methoden ist elementar, da alle Faktoren ineinandergreifen. Auch sind die Entwicklungszeiten oft von diesen Faktoren betroffen, da reale Parameter häufig erst am Ende des Entwicklungszyklus ermittelt werden.

Die verschiedenen Methoden werden vorgestellt und in Übungen von den Teilnehmern unterstützt umgesetzt. Viele Einflussfaktoren sind zu berücksichtigen: Taktversorgung, Toggleraten, Aktivitätsraten, Spannungspotentiale, HDL Kodierung, Synthese- und Implementierungseinstellungen sowie Umgebungsparameter. Am Anfang des

Entwicklungsprozesses werden Schätzverfahren notwendig sein, bei der HDL Entwicklung Simulationsparameter und am Ende des Entwicklungsprozesses Messwerte. Sie lernen die verschiedenen Verfahren anzuwenden.

Optimieren Sie die Leistung des FPGAs und optimieren Sie Ihre Methodik!

Anwendbare Technologien

XILINX FPGAs, SoC und MPSoC Familien

Voraussetzungen

Grundkenntnisse von FPGA Architekturen
Grundkenntnisse des XILINX FPGA Design Flows

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Introduction to Power Optimization
- FPGA Power Requirements
- XILINX Power Estimator Spreadsheet
- Vivado Power Analysis and Optimization
- Power and Temperature Measurement Features
- Power Management Design Techniques
- Power Optimization of IO Resources
- 7 Series Power Management Features
- UltraScale Architecture Power Management Features
- Zynq UltraScale+ MPSoC Power Management

- Zynq UltraScale+ MPSoC Power Domains
- Introduction to Dynamic Function Exchange
- How to Solve a Power Problem

Übungen

- **Lab:** Power Estimation with XPE
- **Lab:** Power Analysis Using the Vivado IDE
- **Lab:** Dynamic Power Estimation with Vivado Power Analysis

Workshop Dynamic Function eXchange (DFX)

Mit Einführung der XILINX Vivado Tool Suite wurde die Methode der partiellen Rekonfiguration deutlich vereinfacht. Nun wird das Verfahren insbesondere auch in der VITIS Unified Plattform Umgebung erweitert, womit auch eine neue Benennung erfolgt:

Dynamic Function eXchange (DFX). Diese bietet insbesondere auch prozessorbasierte Designs wie mit SoCs, MP-SoCs und ACAPs, dass Hardware-Funktionen zur Laufzeit ausgetauscht werden können.

Teilnehmer lernen in diesem Workshop unter Vivado Projekte zu erstellen, um FPGAs partiell zu konfigurieren: FPGA Funktionseinheiten können in eigenständigen Bitstreams implementiert und auch im laufendem Betrieb partiell nachgeladen werden.

Der Kurs zeigt die Methoden der PR-Entwurfspraxis und die notwendigen Abläufe im Entwicklungsprojekt. Auch werden Techniken erlernt, wie Coding Style Einfluss auf die Ressourcen und Timingbedingungen hat. Die unterschiedlichen Varianten des FPGA-Konfigurierens werden in Theorie und Praxis behandelt. Modulare Konfigurationen können wiederum unter Nutzung eines Embedded

Controller gesteuert erfolgen.

Mit diesem Ansatz lassen sich auch sehr große Designs sinnvoll auf kleinere, günstigere FPGAs implementieren. Zielgruppe sind FPGA Hardware Entwickler mit guten Kenntnisse der FPGA Architektur sowie des Vivado Tool Flows, die mit der partiellen Designmethodik die Möglichkeiten der In-System-Programmierung erweitern möchten.

Anwendbare Technologien

XILINX FPGA, SoCs, MPSoCs, RFSocS und ACAPs

Voraussetzungen

Gute FPGA Grundkenntnisse

Sprachkenntnisse VHDL oder Verilog

Grundkenntnisse des Vivado Tool Flows

Empfohlen: Mikroprozessor Kenntnisse

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Partial Reconfiguration Overview

- Partial Reconfiguration Methodology
- PR Terminology
- PR Design Flow

Partial Reconfiguration Tool Flow

- FPGA Configuration Overview
- Configuration Modes

Partial Reconfiguration Bitstreams

- Bitstreams Definitions
- Bitstream Integrity
- ICAP Silicon Resource

Partial Reconfiguration: Managing Timing

- Timing Constraints
- Timing Analysis

Partial Reconfiguration in Embedded Systems

- Zynq SoC/MPSoC Block Diagram
- Processor Configuration for Partial Reconfiguration

- Dynamic Function eXchange in Vitis

Debugging Partial Reconfiguration Designs

- Debugging in PR
- HDL Simulation
- Using the ICAP Output

Partial Reconfiguration Design Recommendations

- Design Requirements and Guidelines
- Design Recommendations
- PR Tool Flow Recommendations

PCIe Core and Partial Reconfiguration

- Application Example: PCIe Core Fast Load
- Tandem PROM and Tandem PCIe
- Partial Reconfiguration over PCIe in UltraScale Devices

Übungen

- **Labs:** Umfangreiche themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung

Workshop Compact FPGA Schaltungstechnik

Der Workshop zum digitalen Schaltungsentwurf – insbesondere für die FPGA Technologie – ist das Fundament für einen Großteil der Kurse in unserem Schulungsprogramm. In der allgemeinen Einführung zu „Compact FPGA Schaltungstechnik“ werden die grundlegenden Schaltungselemente eines FPGA besprochen. Dies umfasst kombinatorische und sequentielle Grundsaltungen wie Multiplexer, Lookup-Tables, Flip-Flops, RAMs, Addierer, Multiplizierer, Taktgeneratoren und I/Os. Anschließend werden komplexere Grundsaltungen wie Vergleicher, Zähler, Schieberegister, FIFO, Stack und Zustandsautomaten aus Basiselementen aufgebaut. Das so gewonnene Grundwissen wird in einem zweiten Schritt auf die UltraScale+ Technologie von Xilinx übertragen. Um weitere theoretische und praktische Grundlagen zu legen, werden fundierte Kenntnisse zu Taktnetzwerken, der Reset-Verteilung, sowie zu „asynchronous vs. synchronous“ vermittelt. Dies umfasst auch Designs mit mehreren Taktnetzwerken und die Datenübergabe bzw. Synchronisierung zwischen verschiedenen Taktdomänen (Clock-Domain-Crossing – CDC).

Die Problemstellungen des synchronen Schaltungsentwurfs sind nicht nur auf Schaltungen im FPGA beschränkt. Ein weiterer Abschnitt der Schulung widmet sich den Problemen und Lösungen für digitale Ein- und Ausgänge wie auch der zeitlichen Relation zwischen Datenpfaden und Taktverteilungen in einer Schaltung aus mehreren digitalen Bausteinen. Kernpunkte sind hierbei das Verstehen von Propagation-Delays sowie Setup- und Hold-Zeiten.

Anwendbare Technologien

alle Xilinx FPGAs inklusive SoCs

Voraussetzungen

keine

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Generische FPGA Architektur

- Primitive Schaltungselemente des FPGA
- Interne Verdrahtung, Verschaltungsmatrizen
- Dedizierte Arithmetikblöcke und Speicher
- I/O Ressourcen und Taktnetzwerke
- Integrierte IP Blöcke
- SoC (Prozessoren, FPGA, Peripherie)

Grundsaltungen

- Flip-Flop, Register, Schieberegister
- Addierer, Zähler
- RAM, ROM
- FIFO, Stack
- Zustandsautomat (FSM)

Xilinx UltraScale+ Technology

- Logik- und Arithmetikressourcen
- Verdrahtung und Taktnetzwerke (inkl. Clock Buffer und PLLs)
- Speicher und I/O Ressourcen
- Integrierte IP Blöcke (z.B. MGT, PCIe)

Codes und Protokolle

- 1-aus-N-Code, Johnson-Code, Grey-Code
- Manchester, 8b/10b, 128b/130b
- Streaming Protokolle

Übungen

PowerWorkshop Professional FPGA Schaltungstechnik

Der Workshop zum digitalen Schaltungsentwurf – insbesondere für die FPGA Technologie – ist das Fundament für einen Großteil der Kurse in unserem Schulungsprogramm.

In der allgemeinen Einführung zu „Professional FPGA Schaltungstechnik“ werden die grundlegenden Schaltungselemente eines FPGA besprochen. Dies umfasst kombinatorische und sequentielle Grundschaltungen wie Multi-plexer, Lookup-Tables, Flip-Flops, RAMs, Addierer, Multiplizierer, Taktgeneratoren und I/Os. Anschließend werden komplexere Grundschaltungen wie Vergleicher, Zähler, Schieberegister, FIFO, Stack und Zustandsautomaten aus Basiselementen aufgebaut. Das so gewonnene Grundwissen wird in einem zweiten Schritt auf die UltraScale+ Technologie von Xilinx übertragen. Um weitere theoretische und praktische Grundlagen zu legen, werden fundierte Kenntnisse zu Taktnetzwerken, der Reset-Verteilung, sowie zu „asynchronous vs. synchronous“ vermittelt. Dies umfasst auch Designs mit mehreren Taktnetzwerken und die Datenübergabe bzw. Synchronisierung zwischen verschiedenen Taktdomänen (Clock-Domain-Crossing – CDC). Die Problemstellungen des synchronen Schaltungsentwurfs sind nicht nur auf

Schaltungen im FPGA beschränkt. Ein weiterer Abschnitt der Schulung widmet sich den Problemen und Lösungen für digitale Ein- und Ausgänge wie auch der zeitlichen Relation zwischen Datenpfaden und Taktverteilungen in einer Schaltung aus mehreren digitalen Bausteinen. Kernpunkte sind hierbei das Verstehen von Propagation-Delays sowie Setup- und Hold-Zeiten. Im abschließenden Kapitel der Schulung werden Datenaustauschprotokolle zur Kopplung von Grundschaltungen, IP Cores, Teilsystemen oder auch Prozessoren besprochen. Dies umfasst klassische off-Chip Protokolle wie UART, SPI, I²C; zudem Address- und Streaming basierte on-Chip Protokolle wie beispielsweise AXI4, AXI4-Lite, AXI-Stream.

Anwendbare Technologien

Alle Xilinx FPGAs inklusive SoCs

Voraussetzungen

keine

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Generische FPGA Architektur

- Primitive Schaltungselemente des FPGA
- Interne Verdrahtung, Verschaltungsmatrizen
- Dedizierte Arithmetikblöcke und Speicher
- I/O Ressourcen und Taktnetzwerke
- Integrierte IP Blöcke
- SoC (Prozessoren, FPGA, Peripherie)

Grundschaltungen

- Flip-Flop, Register, Schieberegister
- Addierer, Zähler
- RAM, ROM
- FIFO, Stack
- Zustandsautomat (FSM)

Xilinx UltraScale+ Technology

- Logik- und Arithmetikressourcen
- Verdrahtung und Taktnetzwerke (inkl. Clock Buffer und PLLs)
- Speicher und I/O Ressourcen
- Integrierte IP Blöcke (z.B. MGT, PCIe)

Codes und Protokolle

- 1-aus-N-Code, Johnson-Code, Grey-Code
- Manchester, 8b/10b, 128b/130b
- Streaming Protokolle
- Addressbasierte Protokolle

Clock-Domain-Crossing (CDC)

- Metastabilität
- Synchronisierschaltungen

Design-Pattern/Best Practices

- Design Optimierungen
- Coding Rules

Inputs/Outputs

- System vs. Source-Synchronous Schnittstellen
- SDR vs. DDR Interfaces

Übungen

Workshop Compact VHDL for Synthesis

Programmierbare Logikbausteine, wie FPGAs, haben sich in allen Bereichen unseren täglichen Lebens etabliert. Sie werden z.B. in mobilen Telefonen, IoT-Geräten, Automobilen oder Rechenzentren verbaut. Ihre Einsatzbereiche sind dabei so vielfältig wie ihre Größe. Sie dienen dabei als Protokolladapter, als Signalkonverter oder als Beschleuniger für die Video-, Radar und Sensordatenauswertung.

Der Entwurf digitaler Schaltungen dieser Größenordnung bedingt den Einsatz einer leistungsfähigen Hardwarebeschreibungssprache, die durch diverse Abstraktionsmechanismen den Entwickler in die Lage versetzt, schnell und effektiv ein Hardwaredesign zu erstellen. VHDL erfüllt all diese Anforderungen.

VHDL ist eine stark typisierte Hardwarebeschreibungssprache, die bereits während des Entwurfs verschiedenfältige Programmierfehler ausschließt. Typischerweise wird VHDL auf dem Register-Transfer-Level (RTL) eingesetzt um digitale Schaltungen beliebiger Komplexität zu entwerfen. Darüber hinaus kann VHDL zur Integration größerer Teilschaltungen auf Systemebene eingesetzt werden. Neben den VHDL Sprachkonstrukten für die Synthese, bietet die Sprache diverse Funktionalitäten um komplexe Verifikati-

onsmodelle zu beschreiben. Damit ist es möglich digitale Schaltungen vom simplen Gatter bis zum System-on-Chip (SoC) vorab zu verifizieren.

In diesem Workshop werden die Synthese relevanten Aspekte der Hardwarebeschreibungssprache VHDL, basierend auf der aktuellen Sprachrevision IEEE Std. 1076-2008, behandelt. Das theoretische Wissen wird dabei mit ausgewählten Beispielen und Übungen am PC vertieft.

Ein grundlegendes Verständnis der digitalen Schaltungstechnik (Gatter, Multiplexer, Flip-Flop, Speicher), sowie die Konzepte einer beliebigen Programmier- oder Skriptingsprache sind wünschenswert.

Anwendbare Technologien

Alle (unabhängig von der FPGA Technologie)

Voraussetzungen

Grundlegendes Verständnis der digitaler Schaltungstechnik (z.B. Compact FPGA Schaltungstechnik)

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Einführung in VHDL

- Sprachkonzept
- Hardware Modellierungstechniken
- Designflow

Hardwarebeschreibung mit VHDL

- Entity / Architecture
- Configuration
- Package
- Library / Context

Sprachelemente

- Signale, Variablen, Konstanten
- Prozesse und Nebenläufigkeit
- Kontrollstrukturen
- Funktionen / Prozeduren
- Generische Beschreibungen

Starke Typisierung in VHDL

- Vordefinierte Typen
- Nutzerdefinierte Typen
- Vordefinierte Operatoren
- Attribute
- Fixed Point Package

Beschreibungstechniken

- Finite State Machines
- Speicher

Testbench Konzept

- Einfache Testbenches
- Assertions

Übungen

Workshop Compact VHDL for Simulation

Programmierbare Logikbausteine, wie FPGAs, haben sich in allen Bereichen unseren täglichen Lebens etabliert. Sie werden z.B. in mobilen Telefonen, IoT-Geräten, Automobilen oder Rechenzentren verbaut. Ihre Einsatzbereiche sind dabei so vielfältig wie ihre Größe. Sie dienen dabei als Protokolladapter, als Signalkonverter oder als Beschleuniger für die Video-, Radar und Sensordatenauswertung. Der Entwurf digitaler Schaltungen dieser Größenordnung bedingt den Einsatz einer leistungsfähigen Hardwarebeschreibungssprache, die durch diverse Abstraktionsmechanismen den Entwickler in die Lage versetzt, schnell und effektiv ein Hardwaredesign zu erstellen. VHDL erfüllt all diese Anforderungen.

VHDL ist eine stark typisierte Hardwarebeschreibungssprache, die bereits während des Entwurfs verschiedenfältige Programmierfehler ausschließt. Typischerweise wird VHDL auf dem Register-Transfer-Level (RTL) eingesetzt um digitale Schaltungen beliebiger Komplexität zu entwerfen. Darüber hinaus kann VHDL zur Integration größerer Teilschaltungen auf Systemebene eingesetzt werden. Neben den VHDL Sprachkonstrukten für die Synthese, bietet die Sprache diverse Funktionalitäten um komplexe Verifikationsmodelle zu beschreiben. Damit ist es möglich digitale

Schaltungen vom simplen Gatter bis zum System-on-Chip (SoC) vorab zu verifizieren.

In diesem Workshop werden, unter Rückgriff auf bereits erlerntes VHDL Wissen, die erweiterten Sprachkonstrukte zum Erstellen von Simulationen nach dem VHDL Testbench Konzept behandelt. Die vermittelten Sprachelemente der Hardwarebeschreibungssprache VHDL, basierend auf der aktuellen Sprachrevision IEEE Std. 1076-2008. Das theoretische Wissen wird dabei mit ausgewählten Beispielen und Übungen am PC vertieft.

Grundlagen in der Hardwarebeschreibungssprache VHDL, z.B. aus Compact VHDL for Synthesis sind wünschenswert.

Anwendbare Technologien

Alle (unabhängig von der FPGA Technologie)

Voraussetzungen

Grundlagen in der Hardwarebeschreibungssprache VHDL (z.B. Compact VHDL for Synthesis)

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

VHDL Rückblick

- Prozesse und Nebenläufigkeit
- Wait Statements, Delta-Cycles
- Typen, Operatoren, Attribute
- Kontrollstrukturen
- Funktionen / Prozeduren
- Generics

Testbench Konzept

- Inline Testbenches
- Modulare Testbenches
- Code vs. funktionalem Coverage
- Assertions

Test Planung

- Directed Testing
- Random Testing

Stimuli und Checks

- Einfache Stimuli Erzeugung
- Analoge Stimuli
- Zufällige Stimuli
- Timing Checks
- Self-Checking Testbench

File I/O

- Lesen und schreiben von Dateien
- Logging

Modellierung externer Komponenten

- Fallbeispiel: UART

Übungen

PowerWorkshop Professional VHDL

Programmierbare Logikbausteine, wie FPGAs, haben sich in allen Bereichen unseren täglichen Lebens etabliert. Sie werden z.B. in mobilen Telefonen, IoT-Geräten, Automobilen oder Rechenzentren eingesetzt. Ihre Einsatzbereiche sind dabei so vielfältig wie ihre Größe. Sie dienen dabei als Protokolladapter, als Signalkonverter oder als Beschleuniger für die Video-, Radar und Sensordatenauswertung. Der Entwurf digitaler Schaltungen dieser Größenordnung bedingt den Einsatz einer leistungsfähigen Hardwarebeschreibungssprache, die durch diverse Abstraktionsmechanismen den Entwickler in die Lage versetzt, schnell und effektiv ein Hardwaredesign zu erstellen. VHDL erfüllt all diese Anforderungen.

VHDL ist eine stark typisierte Hardwarebeschreibungssprache, die bereits während des Entwurfs verschiedenfältige Programmierfehler ausschließt. Typischerweise wird VHDL auf dem Register-Transfer-Level (RTL) eingesetzt um digitale Schaltungen beliebiger Komplexität zu entwerfen. Darüber hinaus kann VHDL zur Integration größerer Teilschaltungen auf Systemebene eingesetzt werden. Neben den VHDL Sprachkonstrukten für die Synthese, bietet die Sprache diverse Funktionalitäten, um komplexe Verifika-

tionsmodelle zu beschreiben. Damit ist es möglich digitale Schaltungen vom simplen Gatter bis zum System-on-Chip (SoC) vorab zu verifizieren.

In diesem Workshop werden alle Aspekte der Hardwarebeschreibungssprache VHDL, basierend auf der aktuellen Sprachrevision IEEE Std. 1076-2008, behandelt. Das theoretische Wissen wird dabei mit ausgewählten Beispielen und Übungen am PC vertieft.

Ein grundlegendes Verständnis der digitalen Schaltungstechnik (Gatter, Multiplexer, Flip-Flop, Speicher), sowie die Konzepte einer beliebigen Programmier- oder Skriptingsprache sind wünschenswert.

Anwendbare Technologien

Alle (unabhängig von der FPGA Technologie)

Voraussetzungen

Grundlegendes Verständnis der digitaler Schaltungstechnik (z.B. Compact FPGA Schaltungstechnik)

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Generische FPGA Architektur

- Schaltungselemente in einem FPGA
- Dedizierter Arithmetikblöcke und Speicher
- I/O Ressourcen und Taktnetzwerke
- Integrierte IP Blöcke

Einführung in VHDL

- Sprachkonzept
- Hardware Modellierungstechniken
- Designflow

Hardwarebeschreibung mit VHDL

- Entity / Architecture
- Configuration
- Package
- Library / Context

Testbench Konzept

- Einfache Testbenches
- Assertions

Sprachelemente

- Signale, Variablen, Konstanten
- Prozesse und Nebenläufigkeit
- Kontrollstrukturen

- Funktionen / Prozeduren
- Generische Beschreibungen

Starke Typisierung in VHDL

- Vordefinierte Typen und Operatoren
- Nutzerdefinierte Typen
- Attribute
- Fixed Point Package

Beschreibungstechniken

- Finite State Machines
- Speicher

Stimuli und Checks

- Einfache Stimuli Erzeugung
- Analoge und zufällige Stimuli
- Timing Checks
- Self-Checking Testbench

File I/O

- Lesen und schreiben von Dateien
- Logging

Übungen

Workshop Compact VHDL Testbenches and Verification with OSVVM

Heutige FPGA und ASIC Designs haben sich in Größe und Komplexität drastisch seit den Anfängen des digitalen Schaltungsentwurfs weiterentwickelt. Diese aufwendigen Schaltungen werden daher mit Hardware-beschreibungssprachen, wie VHDL, als Hierarchie von Teilsystemen erstellt. Die Teilsysteme werden dabei meist durch standardisierte Businfrastrukturen wie AXI, PLB, Avalon oder WishBone verbunden, sowie mit einem Soft-CPU-IP-Core oder einem eingebettetem ARM Prozessor kombiniert. Solch ein Entwurf ist bei weitem zu komplex als das man ihn mit simplen Assertion-basierten VHDL Testbenches verifizieren könnte.

Mit der Open Source VHDL Verification Methodology (OSVVM) wird eine strukturierte Vorgehensweise aufgezeigt, welche eine hohes Maß an Wiederverwendbarkeit im Testbench Code ermöglicht. OSVVM ist eine freie und als Open-Source verfügbare VHDL Library, die Pakete, Datentypen und Unterprogramme, sowie Algorithmen anbietet, welche in fast jeder Testbench benötigt werden. Es besteht kein Grund das Rad ständig neu zu erfinden. Als neuestes Feature bietet OSVVM nun auch vordefinierte Verifikations-IPs, sodass ein breites Spektrum an Standardbussen abgedeckt wird.

OSVVM bietet eine Methodik, welche die folgenden Themenschwerpunkte abdeckt: Transaction-Based Modeling (TBM), Self-Checking, Scoreboards, Memory Modeling, Functional Coverage, Directed, Algorithmic and Constrained Random, sowie Intelligent Testbench Test Generation. Eine VHDL Testbench-Umgebung basierend auf OSVVM ist dabei so mächtig wie andere vergleichbare Verifikationssprachen, wie beispielsweise SystemVerilog oder e'.

Die Schulung startet mit einfachen Testbenches und steigert den Abstraktionsgrad kontinuierlich. Die Teilnehmer lernen die Verwendung von Unterprogrammen und Bibliotheken, Lesen und Schreiben von Dateien, Herausforderungen bei der Modellierung, Transactions-Based Testbenches, Bus Functional Models (BFM), Transaction Basen Models (TBM), Record Datentypen, Resolution Functions, Abstraktionen für Interface Connectivity, Methoden zur Modellsynchronisierung, Protected Types, Access Types (Pointer), unterschiedliche Datenstrukturen (z.B. Scoreboards), Directed, Algorithmic, Constrained Random und Coverage Driven Random Testzeugung, Self-Checking (Ergebnisse, Timing, Protokoll-Tests und Error Injection), Functional Coverage, Darstellung von Analogwerten und periodischen Signalverläufen, Timing und Ausführung des Codes, Testpläne und Configurations kennen.

Dieser 3-Tages Kurs konzentriert sich auf die Theorie, die notwendig ist, um leistungsfähige VHDL Testbench Umgebungen zu entwickeln. Die praktischen Übungen werden aufgrund der kurzen Kursdauer auf das Nötigste beschränkt.

Anwendbare Technologien

Alle (unabhängig von FPGA Technologie)

Voraussetzungen

Fundierte Kenntnisse in VHDL und digitalem Schaltungsentwurf (z.B. Professional VHDL)

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- From Basics to subprograms
- Transaction-based models (TBM / BFM)
- Elements of a transaction-based model
- Data structures for verification
- Creating tests
- Constrained random testing
- Functional coverage
- Execution and timing
- Advanced coverage
- Advanced randomization
- Test plans

PowerWorkshop Professional VHDL Testbenches and Verification with OSVVM

Heutige FPGA und ASIC Designs haben sich in Größe und Komplexität drastisch seit den Anfängen des digitalen Schaltungsentwurfs weiterentwickelt. Diese aufwendigen Schaltungen werden daher mit Hardwarebeschreibungssprachen, wie VHDL, als Hierarchie von Teilsystemen erstellt. Die Teilsysteme werden dabei meist durch standardisierte Businfrastrukturen wie AXI, PLB, Avalon oder WishBone verbunden, sowie mit einem Soft-CPU-IP-Core oder einem eingebettetem ARM Prozessor kombiniert. Solch ein Entwurf ist bei weitem zu komplex, als dass man ihn mit simplen Assertion-basierten VHDL Testbenches verifizieren könnte.

Mit der Open Source VHDL Verification Methodology (OSVVM) wird eine strukturierte Vorgehensweise aufgezeigt, welche ein hohes Maß an Wiederbenutzbarkeit im Testbench Code ermöglicht. OSVVM ist eine freie und als Open-Source verfügbare VHDL Library, die Pakete, Datentypen und Unterprogramme, sowie Algorithmen anbietet, welche in fast jeder Testbench benötigt werden. Es besteht kein Grund das Rad ständig neu zu erfinden. Als neuestes Feature bietet OSVVM nun auch vordefinierte Verifikations-IPs, sodass ein breites Spektrum an Standardbussen abgedeckt wird.

OSVVM bietet eine Methodik, welche die folgenden Themenschwerpunkte abdeckt: Transaction-Based Modeling (TBM), Self-Checking, Scoreboards, Memory Modeling, Functional Coverage, Directed, Algorithmic and Constrained Random, sowie Intelligent Testbench Test Generation. Eine VHDL Testbench-Umgebung basierend auf OSVVM ist dabei so mächtig wie andere vergleichbare Verifikationssprachen, wie beispielsweise SystemVerilog oder ‘c’.

Die Schulung startet mit einfachen Testbenches und stei-

gert den Abstraktionsgrad kontinuierlich. Die Teilnehmer lernen die Verwendung von Unterprogrammen und Bibliotheken, Lesen und Schreiben von Dateien, Herausforderungen bei der Modellierung, Transactions-Based Testbenches, Bus Functional Models (BFM), Transaction Based Models (TBM), Record Datentypen, Resolution Functions, Abstraktionen für Interface Connectivity, Methoden zur Modellsynchronisierung, Protected Types, Access Types (Pointer), unterschiedliche Datenstrukturen (z.B. Scoreboards), Directed, Algorithmic, Constrained Random und Coverage Driven Random Testerzeugung, Self-Checking (Ergebnisse, Timing, Protokoll-Tests und Error Injection), Functional Coverage, Darstellung von Analogwerten und periodischen Signalverläufen, Timing und Ausführung des Codes, Testpläne und Configurations kennen. Abschließend wird ein TBM für AXI4-Stream Master und Slaves untersucht.

Dieser Kurs enthält mehrere Beispiele, die direkt als Vorlage bei der Entwicklung eigener Testbenches benutzt werden können. Als Ergebnis erhalten Sie eine Testumgebung auf Systemebene, die transaktionsgesteuert und selbsttestend ist. Praktische Übungen bieten die Möglichkeit das Gelernte anzuwenden.

Anwendbare Technologien

Alle (unabhängig von FPGA Technologie)

Voraussetzungen

Fundierte Kenntnisse in VHDL und digitalem Schaltungsentwurf (z.B. Professioneller VHDL)

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Testbench overview
- From basics to subprograms
- Transactions and subprograms
- Modeling for verification
- VHDL I/O
- **Lab Review:** Testing w/ subprograms
- Transaction-based models (TBM / BFM)
- Elements of a transaction-based model
- Data structures for serification
- **Lab Review:** UartTx BFM
- Creating sestis
- Constrained random testing
- Functional coverage
- Execution and timing
- Configurations and simulation management
- Advanced coverage
- Advanced randomization
- **Lab Review:** Scoreboards, randomization and coverage
- Test plans
- Modeling RAM
- Transaction-based BFM part 2
- AXI4-Stream Master and Slave

Workshop Compact Verilog

Durch den stetig steigenden Bedarf nach programmierbaren Bausteinen hoher Komplexität verbunden mit hohen Taktraten werden die Anwender mit neuen, ebenfalls stetig wachsenden Herausforderungen konfrontiert. Da Schnelligkeit, Flexibilität und hohes Qualitätsniveau entscheidend für den wirtschaftlichen Erfolg sind, müssen moderne und leistungsfähige, auf Hardware-Beschreibungssprachen basierende Designmethoden eingesetzt werden, um der Forderung nach kürzesten Entwicklungszeiten und hoher Qualität gerecht zu werden.

Der PLC2-Workshop „Compact Verilog“ schult den Anwender im Einsatz von Verilog HDL. Die Teilnehmer lernen die grundlegenden Beschreibungselemente sowie die Einsatzbereiche von Verilog kennen. Die theoretischen

Inhalte werden durch Übungen am PC abgerundet. Zur Verilog Synthese und Simulation arbeiten dabei bis zu maximal zwei Teilnehmer gemeinsam an einem PC und Evaluierungsboard.

Anwendbare Technologien

alle (von Technologie unabhängig)

Voraussetzungen

Grundkenntnisse Digitaltechnik (wie z.B. in „Compact FPGA Schaltungstechnik“ vermittelt) sind hilfreich

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Hardware Modeling Overview

- History and Formalization of Verilog
- Levels of Abstraction
- Inference and Instantiation

Verilog Language Concepts

- Keywords and Identifiers
- Signals and Data Types
- Numerical Representation
- Arrays

Memories, Modules and Ports

- Port Rules and Restrictions
- Creating Hierarchical Modules
- Modeling RAM and ROM

Introduction to Testbenches

- The Testbench Concept
- Behavioral Coding

Operators and Expressions

- Using Operators
- Signed and Unsigned Objects

Data Flow-Level Modeling

- Continuous Assignment
- Delay Specifications

Tasks and Functions

- Subprograms, Tasks, Functions
- Header Files

Verilog Procedural Statements

- Differences Between RTL and Behavioral Coding
- Procedural Statements
- Blocking and Non-Blocking Statements

Controlled Operation Statements

- if/else Statements
- case Statements
- Loop Statements

Advanced Language Concepts

- Event-Driven Simulation
- Using wait and Local Parameters
- Verilog Generate Statements

Finite State Machines

- FSM Overview
- Encoding Style
- Single-Process or Multi-Process FSMs
- FSM Example

Targeting XILINX FPGAs

- Timing and Performance Guidelines
- Synthesis Tools and Options

Advanced Verilog Testbenches

- Verilog File I/O
- Read Functions
- Write System Tasks

Workshop SystemVerilog – Advanced Verification for FPGA Design

FPGA Designs haben sich in Bezug auf Performance und Umfang stark weiterentwickelt. Die Verifikation dieser Art von Designs sind zu einer enormen Aufgabe herangewachsen, besonders im Zusammenhang mit der Validierung gegen den Testplan und die Spezifikation. SystemVerilog stellt eine umfangreiche Menge von Werkzeugen zur Verfügung und ist eine natürliche Erweiterung von Verilog. Die Sprache kennt auch Konstrukte, die besser die Intension widerspiegeln, wie z.B. Aufzählungstypen, integrierte Assertions und Hochsprachenkonstrukte, die sowohl strukturelle Hierarchie als auch OOP unterstützen und erlaubt auch, flexiblere und wiederverwendbare Testbenchkomponenten zu erstellen, auch im Kontext von VHDL Designs.

Dieser Workshop wird einen Überblick über die Sprache SystemVerilog geben und neue Methodiken wie Assertion-Based Verification, Constrained Randomization und Functional Coverage vorstellen. Die Teilnehmer werden lernen, wie diese Methodiken zur Verifikation eines VHDL Designs genutzt werden können, um die Verifikation zu beschleunigen und

den aktuellen Stand bezogen auf das Ziel, den Nachweis der Verifikation aller Funktionen, zu erbringen.

Zielgruppe

FPGA Design- und Verifikationsingenieure

Voraussetzungen

Erfahrung mit VHDL oder Verilog für Design und Verifikation

Ziele

Grundlagen der Sprache SystemVerilog
Einführung in das Konzept des OOP (Objekt orientierte Programmierung) in SystemVerilog
Einführung in SystemVerilog Assertions, Constrained Randomization und Functional Coverage und deren Einbindung in eine Testbench für ein VHDL Design

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Motivation

Einführung in SystemVerilog

- Datentypen
- Felder und Strukturen
- Ablaufsteuerung
- Hierarchie
- Tasks, Funktionen, dynamische Prozesse
- Klassen in OOP

SystemVerilog Assertions

- SVA Sprachaufbau
- Sequenzielle Ausdrücke
- Property Ausdrücke
- Assert und Cover Direktive
- Bind Statement

Constrained Randomization

- Constraints
- Random Case
- Random Sequence

Functional Coverage

- Covergroup
- Coverpoint
- Cross

Workshop UVM Made Easy for FPGA Designers

Heutige FPGA Designs haben sich zu SoC Designs hin entwickelt, so dass deren Komplexität einen Grad erreicht hat, der vor einigen Jahren noch für ASIC Designs charakteristisch war. Die Aufgabe, für die vollständige Verifikation eines solchen Systems Testbenches zu schreiben, ist zu einer großen Herausforderung geworden. Unterschiedliche Verifikationsmethoden haben dies auf unterschiedliche Arten adressiert. Der umfassendste Ansatz ist die Universal Verification Methodology, UVM. Diese wurde nun auch zum Standard (IEEE 1800.2-2017). Sie stellt SystemVerilog Basisverifikationskomponenten zur Verfügung, mit denen eine Testbench-Infrastruktur erstellt werden kann, die sehr stark die Wiederverwendbarkeit unterstützt.

Wegen der Komplexität der UVM Bibliothek ist das Erstellen einer Testbench eine zeitaufwändige Aufgabe und erfordert umfassendes Wissen über die Möglichkeiten, die die Bibliothek bietet. Um die Verifikationsingenieure beim initialen Erstellen einer Testbench-Infrastruktur zu unterstützen, wurde das UVM-Framework entwickelt. Mit diesem kann eine UVM Testbench sehr schnell erstellt werden und durch Änderungen an wenigen Stellen wird diese innerhalb weniger Stunden simulierbar.

Der Workshop „UVM Made Easy for FPGA designers“ wird die wichtigsten UVM Bausteine vorstellen, und damit die Grundlagen für das Funktionieren einer UVM Testbench,

den Prozess der Instanzbildung und die Kommunikation zwischen den UVM Komponenten und dem DUT, vermitteln. Darauf aufbauend werden die UVM Framework Verifikationsbausteine und das auf Python basierende API.

Die Teilnehmer werden in dem Übungsteil eine UVM Testbench mit dem UVM Framework für ein einfaches Beispiel-Design in mehreren Schritten erstellen. Am Ende des Workshops werden die Teilnehmer in der Lage sein, das UVM Framework API zu nutzen, um eine Testbench-Infrastruktur für ihr eigenes FPGA Design zu generieren.

Zielgruppe

FPGA Design- und Verifikationsingenieure

Voraussetzungen

Grundkenntnisse von SystemVerilog und OOP (Objekt orientierte Programmierung) sind von Vorteil

Ziele

Verifikation – Ansätze und Methodiken
UVM – Grundlagen und Prinzipien
UVM Framework – Bibliothekskomponenten, Struktur und API

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction

UVM

- UVM Verification Components
- UVM Transaction Interfaces
- LAB: UVM Transaction Interfaces
- UVM Factory
- LAB: UVM Factory
- UVM configuration database

UVM Framework

- UVMF Base Classes
- UVMF Base Class Package
- Introducing the UVMF API

- Practical Example: Creation of a UVM Testbench
- Introducing the Interface API
- LAB: Generating the FPU interface
- Introducing the Environment API
- LAB: Generating the FPU Environment
- Introducing the Testbench API
- LAB: Generating the FPU Testbench
- LAB: Completing the Testbench
- LAB: Adding another Test
- Conclusion

References

Contact Information

Workshop Developing for Mission Critical FPGA & SoC

FPGA's and heterogeneous SoC's are used in an increasing number of mission critical or high reliability applications. These applications span a diverse range from industrial, medical and scientific to defences, transportation and even space. For these devices to safely and reliability operate in an often-harsh environment, a more rigorous design approach is required. One that introduces both stricter engineering governance in the design process and design mitigation techniques

As such designing these solutions requires the designer to not only understand what techniques can be used at the logic level but also, the wider systematic, regulatory and environmental issues.

This course will therefore present the environmental challenges and what they mean to the logic designer. Along with introducing high level concepts such as SIL level, Reliability and Mean Time to Failure, attendees will also gain an understanding of the importance of engineering governance.

The focus of this course is the development of techniques which can be used in programmable logic including Clocking & Reset strategy, Triple Modular Redundancy, IO

Planning, Safe State Machines and Counters, Error Correcting Codes, Single Event Effect Mitigation along with Verification strategies and metrics, formal equivalence checking, Synthesis strategies and several other advanced techniques. Each session will complete with a Lab which will demonstrate the concepts outlined in the session. Attendees will at the completion of the course have a detailed understanding of the challenges and strategies to address the creation of mission critical systems for a wide range of applications.

Applicable technologies

These techniques outlined in this course can be applied to any FPGA technology. For reference course XILINX Seven Series devices will be targeted.

Requirements

It is expected that the attendee is an experienced FPGA designer and has familiarity with electronics and system engineering concepts.

Duration and Costs:

Duration: 3 days

Cost: € 2,100, – net per person, including detailed training material, drinks in the breaks and lunch.

Agenda

How the environment impacts our designs

- Temperature, Shock & Vibration, EMC and Radiation

Programmatic / System level considerations.

- Different Standards 61508 / DO254 / ISO 26262
- The design life cycles
- Engineering Governance
- What is reliability & What does MTBF Mean & What impacts the MTBF
- Requirement capture & Progressive Verification
- Architectural design & Inter dependency of faults between SW and HW
- Common Cause Failures & Failure Mode & Redundancy
- Worse Case Analysis

FPGA Design Considerations

- FPGA Development overview & Supporting Documentation
- Device Selection – OTP, FLASH, SRAM
- Coding Style & Certified tools
- Failure modes
- Self-Test and Diagnostics
- Clocks and Rest & IO Planning & JTAG / Boundary Scan
- Safer State Machines & Counters
- Error Correcting Codes Communications and Memories
- Triple Modular Redundancy Local, Fine Gain and Global
- Functional Separation within the device, Isolation flow
- Single Event Effects and Configuration Corruption
- Verification & Verification Metrics
- Fault Injection
- Timing Closure
- Advanced Features e.g. XADC, SysMon

Workshop Git

Agile und gemeinschaftliche Softwareentwicklungsprozesse gewinnen zunehmend an Popularität, da sie häufigere Builds, Tests und Integrationen ermöglichen. Sie ermöglichen darüber hinaus eine schnellere Auslieferung und Bereitstellung des Produktes. Da sich die Codebasis zu jeder Zeit in einem „release at anytime“ Zustand befindet, kann die Zeit von der Idee oder dem Problem zur Auslieferung drastisch reduziert werden.

Hardware- und eingebettete Systementwickler mit einem Softwareentwickler Hintergrund kennen die Vorzüge agiler Entwicklungsstile und würden diesen gerne im Bereich der Hardwareentwicklung einsetzen. Leider werden diese Prozesse von EDA Tools nur unzureichend unterstützt. Dieser Workshop wird sich der Herausforderung annehmen einen agilen und teambezogenen Entwicklungsprozess aufzuzeigen der sich für kleine und mittelgroße Hardware- und/oder eingebettete Systeme Entwicklungsteams eignet.

Der Workshop widmet sich drei Kernaufgaben: Den Teilnehmern wird ein tiefgreifendes Verständnis von Version Kontrollsystem und Git im speziellen vermittelt, sodass Git selbstbewusst eingesetzt werden kann. Die Teilnehmer werden in der Lage sein in einem agilen, teamorien-

tierten Entwicklungsprozess mitzuwirken. Darüber hinaus können Sie aus den vermittelten Grundlagen einen angepassten Entwicklungsprozess für das eigene Entwicklungsteam und dessen Besonderheiten ableiten. Abschließend werden Techniken zur Produktivitäts- und Qualitätssteigerung, wie Code Review besprochen.

Alle im Workshop eingesetzten Programme und Cloud-dienste sind seit geraumer Zeit im produktivem Einsatz und werden stetig weiter entwickelt. Des Weiteren können die vorgestellten Clouddienste in einem gesicherten Intranet betrieben werden, welches von einer firmeneigenen IT-Abteilung überwacht wird. Dies ermöglicht den Entwicklungsteam alle brandneuen Features zu nutzen ohne gegen Sicherheitsauflagen zu verstoßen.

Anwendbare Technologien

Alle (unabhängig von FPGA Technologie)

Voraussetzungen

Keine

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Vision: Git verstehen und daraus einen agilen, gemeinschaftlichen Entwicklungsprozess ableiten.

Git Basics

- Einführung in Version Control Systems
- Git Operationen in der Shell und GUI
- Git Referenzen: Branches, Tags, Remotes
- Branching, Merging

GitLab

- Einführung in GitLab
- Issues, Tags, Milestones, Issue Boards
- Merge Requests (Pull Requests)
- Code Review
- GitLab Integrations (Erweiterungen)

Advanced Git

- Branching Modelle und Workflows
- Submodules
- Scripting mit Git

Continuous Workflows

- Continuous Building
- Continuous Integration
- Continuous Delivery / Deployment
- Continuous Documentation

Git and EDA Tools

- XILINX ISE, Vivado
- Aldec ActiveHDL, RivieraPRO
- Mentor Graphics ModelSim, QuestaSim

Labs:

- Git in der Shell und GUI
- Teamwork mit GitLab
- Scripting mit Git

PowerWorkshop Continuous Integration

Agile und gemeinschaftliche Softwareentwicklungsprozesse gewinnen zunehmend an Popularität, da sie häufigere Builds, Tests und Integrationen ermöglichen. Sie ermöglichen darüber hinaus eine schnellere Auslieferung und Bereitstellung des Produktes. Da sich die Codebasis zu jeder Zeit in einem „release-at-anytime“ Zustand befindet, kann die Zeit von der Idee oder dem Problem zur Auslieferung drastisch reduziert werden.

Hardware- und eingebettete Systementwickler mit einem Softwareentwickler Hintergrund kennen die Vorzüge agiler Entwicklungsstile und würden diesen gerne im Bereich der Hardwareentwicklung einsetzen. Leider werden diese Prozesse von EDA Tools nur unzureichend unterstützt. Dieser Workshop wird sich der Herausforderung annehmen einen agilen und teambezogenen Entwicklungsprozess aufzuzeigen der sich für kleine und mittel-große Entwicklungsteams eignet.

Der Workshop widmet sich drei Kernaufgaben: Den Teilnehmern wird ein tiefgreifendes Verständnis von Version Kontrollsystemen und Git im speziellen vermittelt, sodass Git selbstbewusst eingesetzt werden kann. Die Teilnehmer werden in der Lage sein in einem agilen, teamorientierten Entwicklungsprozess mitzuwirken. Darüber hinaus können Sie aus den vermittelten Grundlagen einen angepassten Entwicklungsprozess für das eigene Entwicklungsteam ableiten. Dies beinhaltet auch Techniken zur Produktivitäts- und Qualitätssteigerung. Abschließend werden VUnit und OSVVM Verification IPs besprochen

um VHDL Testbenches in ein vollautomatisiertes Regressionstestsystem zu überführen, welches sich in den Git basierten Arbeitsablauf integriert.

Alle im Workshop eingesetzten Programme und Dienste sind seit geraumer Zeit im produktivem Einsatz und werden stetig weiter entwickelt. Die vorgestellten Cloud-Dienste können in einem gesicherten Intranet betrieben werden, welches von einer firmeneigenen IT-Abteilung überwacht wird. Dies ermöglicht den Entwicklungsteams alle brandneuen Features zu nutzen ohne gegen Sicherheitsauflagen zu verstoßen.

Die Hardwarebeschreibungssprache VHDL und das Erstellen von Assertion basierten Testbenches sind nicht Teil des Kurses. Die Teilnehmer sollten sicher im Umgang mit VHDL-2008 Testbenches und VHDL Simulatoren sein.

Anwendbare Technologien

Alle (unabhängig von FPGA Technologie)

Voraussetzungen

Fundierte Kenntnisse in VHDL und automatisierten VHDL Tests (z.B. aus Adv. VHDL Verification)

Basiswissen in Python oder einer vergleichbaren Scriptingsprache

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Git Basics

- Einführung in Version Control Systems
- Git Operationen in der Shell und GUI
- Git Referenzen: Branches, Tags, Remotes
- Branching, Merging

GitLab

- Issues, Tags, Milestones, Issue Boards
- Merge Requests (Pull Requests)
- Code Review

Advanced Git

- Branching Modelle und Workflows
- Submodules
- Scripting mit Git
- OSVVM Verification IPs

- Einführung in OSVVM
- Einsatz von OSVVM Verification IPs (VIPs)

VHDL Unit Testing mit VUnit

- Einführung in VUnit und TDD
- Umschreiben von VHDL Tests zu VUnit Tests

Continuous Workflows

- Continuous Building / Integration
- Continuous Delivery / Deployment
- Continuous Documentation

Labs:

- Git in der Shell und GUI
- Teamwork mit GitLab
- Benutzung von OSVVM Verification IPs
- Umschreiben von VHDL Tests zu VUnit Tests



Workshop **VIVADO** Design Suite Tool Flow

Bei der Vivado™ Design Suite handelt es sich um eine stark integrierte Entwicklungsumgebung, die auf einer systemorientierten Generation von Werkzeugen basiert. Das Rückgrat hierfür bilden eine übergreifende, skalierbare Datenbasis und eine einheitliche Testumgebung. Gleichzeitig führt die verstärkte Verwendung von Industriestandards wie z.B. AMBA® AXI4, IP-XACT (für Metadaten von eigenen IP-Zellen), Tool Command Language (Tcl), Synopsys® Design Constraints (SDC) etc. zu einer einfachen Erweiterbarkeit oder zur vereinfachten Automatisierbarkeit des Entwicklungsablaufs. Vivado™ ist so konzipiert, dass damit alle Aspekte (Logik, SW, I/O, Mixed Signal, etc.) der programmierbaren Technologien verwaltet werden können und das für Designs bis zu einer Größe von 100M ASIC-Gattern.

Diese Schulung richtet sich an Einsteiger. Sie vermittelt verschiedene Softwarefunktionen sowie Planungstechniken und –strategien. Der Kurs führt den Teilnehmer durch einen vollständigen Designablauf: Von der Spezifikation der Verhaltensweise, über die Synthese, Verifikation und Implementierung bis zum Hardwaretest.

niken und –strategien. Der Kurs führt den Teilnehmer durch einen vollständigen Designablauf: Von der Spezifikation der Verhaltensweise, über die Synthese, Verifikation und Implementierung bis zum Hardwaretest.

Anwendbare Technologien

7 Series FPGAs und neuere

Voraussetzungen

Grundkenntnisse VHDL oder Verilog

Grundkenntnisse Digitaltechnik

Dauer und Kosten

1 Tag, € 750,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Project Planning

- Overview
- Specifications and Design Documents
- Technology Selection
- Verification

Vivado IDE Overview and Projects

- Vivado Tool Features and Benefits
- Creating New Projects and Modifying Projects
- RTL Elaboration
- Simulating the Design
- Synthesis and Implementation
- Configuration

Vivado Tool Flow

- Overview
- Synthesis and Reports
- Implementation and Reports
- Vivado IDE Projects
- Scripted Design Flow

I/O Planning and Clock Constraints

- Overview
- Using the I/O Planner View
- Entering Clock Constraints

Vivado Simulator (XSIM)

- Overview
- HDL Testbenches
- Simulating a Design

Labs based on Soft- and Hardware



Workshop VIVADO Design Suite Static Timing Analysis and XILINX Design Constraints

Bei der Vivado™ Design Suite handelt es sich um eine stark integrierte Entwicklungsumgebung, die auf einer systemorientierten Generation von Werkzeugen basiert. Das Rückgrat hierfür bilden eine übergreifende, skalierbare Datenbasis und eine einheitliche Testumgebung. Gleichzeitig führt die verstärkte Verwendung von Industriestandards wie z.B. AMBA® AXI4, IP-XACT, Tool Command Language (Tcl), Synopsys® Design Constraints (SDC) etc. zu einer einfachen Erweiterbarkeit oder zur vereinfachten Automatisierbarkeit des Entwicklungsablaufs. Dieser Kurs bespricht detailliert die Erstellung von XDC-Constraints und die Durchführung der statischen Timinganalyse. Des Weiteren kommen die korrekte Verwendung der FPGA-Ressourcen und die effektive Nutzung der Vivado™ Design-Datenbasis für z.B. Analysezwecke

zur Sprache. Zusätzlich werden diverse FPGA-spezifische Design-Tipps und -Strategien besprochen.

Anwendbare Technologien

7 Series und UltraScale FPGAs

Voraussetzungen

Kenntnis der FPGA-Technologie, des Vivado Software Flows sowie grundlegendes Constraining
Kenntnisse in VHDL oder Verilog
Solide Kenntnisse in digitaler Schaltungstechnik

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Review of Essentials of FPGA Design

FPGA Design Techniques

Accessing the Design Database

- Finding Objects
- Object Properties
- Object Connectivity
- Using the GUI

Static Timing Analysis and Clocks

- Setup Checks and Clocks
- Timing Reports
- Hold Checks
- Generated Clocks
- Multiple Clocks

Inputs and Outputs

- Creating Input Delays
- Creating Output Delays
- Using Virtual Clocks

Timing Exceptions

- Multicycle Paths
- False Paths
- Max/Min Delay Exceptions
- Constraint Priority

FPGA Design Methodology

- Device and System Architecture
- Pin Planning and Floorplanning Methodology

HDL Coding Techniques

Reset Methodology

- Reset Recommendations
- Reset Workarounds

Synchronization Circuits and the Clock Interaction Report

- Single Bit Synchronization Circuits
- Bus Synchronization Circuits
- Clocks and Synchronizers

Timing Closure

- Baselining
- Isolating Common Timing Bottlenecks
- Last Mile Strategies

FPGA Design Methodology Case Study

Labs



PowerWorkshop Professional VIVADO

Bei der Vivado™ Design Suite handelt es sich um eine stark integrierte Entwicklungsumgebung, die auf einer systemorientierten Generation von Werkzeugen basiert. Das Rückgrat hierfür bilden eine übergreifende, skalierbare Datenbasis und eine einheitliche Testumgebung. Gleichzeitig führt die verstärkte Verwendung von Industriestandards wie z.B. AMBA® AXI4, IP-XACT (für Metadaten von eigenen IP-Zellen), Tool Command Language (Tcl), Synopsys® Design Constraints (SDC) etc. zu einer einfachen Erweiterbarkeit oder zur vereinfachten Automatisierbarkeit des Entwicklungsablaufs. Vivado™ ist so konzipiert, dass damit alle Aspekte (Logik, SW, I/O, Mixed Signal, etc.) der programmierbaren Technologien verwaltet werden können und das für Designs bis zu einer Größe von 100M ASIC-Gattern.

Diese Schulung stellt einen umfassenden Einstieg in diese Entwicklungsumgebung dar. Das Spektrum reicht von

ersten Schritten über fortschrittlichere Techniken bis zum Constraining mit XDC und der damit verbundenen Timing-Analyse.

Anwendbare Technologien

7 Series FPGAs, UltraScale und UltraScale+

Voraussetzungen

Kenntnisse in VHDL oder Verilog

Solide Kenntnisse in digitaler Schaltungstechnik

Kenntnis der FPGA-Technologie

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

FPGA Design Methodology Summary

Introduction to the Vivado Design Suite

Vivado Design Flows

- Vivado IDE Design Flows
- Project-Based Design Flows
- Non-Project Batch Mode

Visualization for Analysis

Designing with IP

- Introduction
- Working with IP
- IP Integrator

Basic Timing Constraints and Reports

- Basic Timing Constraints
- Basic Static Timing Analysis
- Reports

Accessing the Design Database

Static Timing Analysis and Clocks

- Setup Checks and Clocks
- Timing Reports
- Hold Checks
- Generated Clocks
- Multiple Clocks

Input and Output Constraints

Timing Exceptions

- Multicycle Paths
- False Paths
- Max/Min Delay Exceptions

Advanced Timing Analysis

- Floorplan

Advanced I/O Interface Constraints

- Single Data Rate vs. Double Data Rate
- System-Synchronous Interfaces
- Source-Synchronous Interfaces

Project-Based and Non-Project Batch Design Flows

Scripting Using Project Based and Non-Project Batch Flows

FPGA Design Methodology

HDL Coding Techniques

Reset Methodology

Synchronization Circuits and the Clock Interaction Report

Timing Closure

FPGA Design Methodology Case Study

VIVADO® **Workshop** Debugging Techniques Using the VIVADO Logic Analyzer

Mit steigender Integrationsdichte der FPGAs sind dem Entwickler zunehmend weniger Messmöglichkeiten gegeben, das Design in der realen Funktion zu verifizieren.

Die geringe Anzahl von Chip I/Os im Verhältnis zu internen Verbindungen, Kosten und Packungsdichte der Leiterkarte sind Hauptkriterien für ein FPGA-integriertes Logik-Analyse Werkzeug.

In der Vivado Tool Suite wurde das auf Chipscope Pro basierte Tool integriert und bietet Logikanalysefähigkeit für interne Signale der FPGA Fabric mit Konfigurationen von Trigger und Datenspeicherung angepasst der Messaufgabe.

Insbesondere auch VHDL Einsteigern bietet dieses Werkzeug eine Alternative, ihr Design ohne umfangreiche HDL Simulation sicher in Betrieb zu setzen.

Für die Synthese besteht die Anforderung, dass durch Zufügen des Cores, Realtime Anforderungen nach wie vor eingehalten werden.

Die Debug Cores werden an realen Designs erzeugt und mit FPGA Evaluierungsboard erarbeitet.

Die Kursteilnehmer lernen, wie die Anwendung der Software als Logic-Analyzer aufgebaut ist und in welcher Weise dem

Entwickler bekannte Funktionen wie Trigger-Menue und Wave Darstellungen zur Verfügung stehen. Insbesondere bei Embedded-Controller Applikationen ist dieses Werkzeug zu empfehlen, da prozessorinterne Bussignale parallel zum Debugging Werkzeug synchronisiert darstellbar sind. Auch bietet diese Art der Verifikation die Möglichkeit, HDL basierte Simulationen zu beschleunigen, indem Daten aus der Hardware aufgezeichnet und der Simulation bereitgestellt werden. In diesem Workshop werden alle wichtigen Möglichkeiten gezeigt, anspruchsvolle FPGA-Verifikationen der Hardware zu meistern.

Anwendbare Technologien

7 Series FPGAs und neuere Technologien

Voraussetzungen

Grundlegende Kenntnisse der XILINX FPGA Architektur
Grundlagenkenntnisse VHDL

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

The Hardware Debug Methodology and How it Works

Inserting the Debug Cores

Instantiating the Debug Cores

Debug Flow in IP Integrator

Triggering and Storage

Visualizing Data – The Analyzer Tool

Tips and Tricks

Time for Timing

Optimization – Area Groups

Case Studies

Scripting

Remote Access

The Vivado Serial I/O Analyzer

Übungen

- **Lab:** Inserting a Debug Core from the Vivado Design Suite
- **Lab:** Adding a Debug Core Using the HDL Instantiation Flow
- **Lab:** Debugging Flow – IPI Block Design
- **Lab:** Tips and Tricks
- **Lab:** VIO TCL Scripting
- **Lab:** Remote Access

Workshop Tcl Schnellstart

Für XILINX/FPGA Entwickler hat die Skript-Sprache Tcl mehrere Anwendungen:

1. Skript-gesteuerte Automatisierung des „Vivado Design-Flows“
2. „High-Level-Control-Logic“ für Linux/ZYNQ-Applikationen
3. Realisierung eigener, kleiner Tools (evtl. auch mit Tk-basiertem GUI)

Voraussetzung ist jeweils ein grundlegendes Verständnis für die Verwendung von Tcl als Skript- Sprache. Nach einer kurzen Einführung in die wesentlichen Konzepte erhalten die Teilnehmer am ersten Tag dieses Workshops eine kompakte Darstellung der Syntax und einen Überblick über wesentliche Bibliotheksfunktionen. Die Theorie wird dabei durch zahlreiche „live“ vorgeführte Demo-Beispiele praktisch untermauert.

Am zweiten Tag (optional abwählbar – siehe Hinweis am Ende) erhalten die Teilnehmer selbst die Möglichkeit, die am ersten Tag vermittelten Inhalte im Rahmen eines geführten Praktikums durch die Bearbeitung einer Reihe „Mikro-Projekten“ praktisch anzuwenden. Letztere werden in den nachfolgend genannten Themenbereichen, die Auswahl kann individuell erfolgen:

- Allgemeiner Einsatz von Tcl als Skriptsprache (Variablen und Ablaufsteuerung, Zeichenketten, Dateiverarbeitung, Aspekte guter Programmstruktur im Hinblick auf Wartbarkeit, Erweiterbarkeit und Testbarkeit, ...).
- Verwendung von Tcl mit Vivado (Umgang mit den spezifischen Kommandos im „Project“- und „Non-Project“-Modus zur „intelligenten“ Automatisierung von Abläufen, ...).
- Tcl für ZYNQ-Applikationen und eigene, von Vivado unabhängige Tools (klassische vs. ereignisgetriebene Architektur, Debug-Tracing, Paketierungsmöglichkeiten, ...).

Hinweis:

Die Zubuchung des zweiten Tages ist speziell dann verzichtbar, wenn in direktem Anschluss der vertiefende Kurs „Advanced Vivado-Tcl-Scripting“ besucht wird, da dieser in ausreichendem Umfang praktische Übungen zum Tcl-Scripting des Vivado Design- Flows anbietet. Besteht andererseits vor allem Interesse am Einsatz von Tcl als Sprache zur Applikations-Entwicklung oder Rapid-Prototyping für Linux/ZYNQ-basierten Embedded-Plattformen oder eigene, kleine Tools (inkl. dem Einsatz von Tk für grafische Oberflächen) empfehlen wir statt dieses („1+1“-tägigen) Workshops den Besuch des dreitägigen Tcl/Tk-Workshop zu erwägen.

Anwendbare Technologien

Erster Tag: technologieunabhängig; *zweiter Tag* (bei Auswahl von Mikro-Projekten zum Vivado- Scripting): 7-Series oder neuere FPGAs

Voraussetzungen

Grundlagen der Programmierung und Basis-Erfahrung in einer höheren Programmiersprache oder Assembler. Um am *zweiten Tag* Vivado-spezifische Mikro-Projekte bearbeiten zu können, sind zusätzlich entsprechende Kenntnisse im GUI-basierten FPGA Design-Flow mit Vivado erforderlich!

Dauer und Kosten

1 oder 2 Tage, 2. Tag optional: Praxisübungen
€ 750,- netto pro Teilnehmer (1 Tag Workshop)
€ 1.500,- netto pro Teilnehmer (2 Tage Workshop)
inklusive Schulungsunterlagen, Pausengetränken und Mittagessen

Agenda

Erster Tag

- Grundlegende Konzepte
- Syntax-Analyse und Quoting
- Einfache Variablen, Listen und Arrays
- Ablaufsteuerung
- Fehlerbehandlung
- Unterprogramme
- Zeichenketten
- Dateiverarbeitung
- Introspektion
- Fortgeschrittene Features (Ausblick)

Zweiter Tag

Intensiv betreutes Praktikum (in Einzelarbeit oder Zweiergruppe) zur Umsetzung von zwei bis drei „Mikro-Projekten“ mit abschließender Diskussion der erarbeiteten Lösungen. Die konkreten Aufgaben können aus einer Liste von Vorschlägen gewählt werden. Gerne gesehen sind dabei auch kleinere (oder sogar größere) Variationen nach eigenen Vorstellungen (oder gemäß aktuell anstehenden Aufgaben).

Workshop Advanced Vivado-Tcl-Scripting

Der Schwerpunkt der Schulung liegt auf der fortschrittlichen Verwendung des VIVADO TCL Design Flows. Die Verwendung eines script-basierten Design Flows ermöglicht beispielsweise die automatisierte Erstellung von Verzeichnissen und Files, sowie die automatisierte Bearbeitung vollständiger Designs. Diese Methodik ermöglicht zusätzlich einen plattform-unabhängigen Design Flow mit gesteigerter Effizienz bei Designänderungen, Nachverfolgungen und Revisionskontrolle.

Nicht zuletzt stehen dem Anwender hier zusätzliche Optionen und Möglichkeiten zur Verfügung die aus dem GUI basierten Design Flow heraus nicht möglich sind.

Zu Beginn lernt der Teilnehmer sowohl die Unterschiede zwischen einem „Project based Flow“ und einem „Non-Project based Flow“ kennen als auch wie VIVADO solche Skripte auch automatisch erstellen kann. Es wird demonstriert wie Tcl Skripte komplexe Abläufe automatisieren können und wie auch andere Werkzeuge in den Flow mit eingebunden werden können.

Nach der Einführung in die Verwendung/Erzeugung der Tcl-Skripte werden grundlegende und oft benötigte Strukturen und Methoden behandelt wie z.B. Listen,

Schleifen, Fehlerbehandlung, String- und Dateiverarbeitung.

Gerade wenn es darum geht komplexe Abläufe zu steuern benötigt man den automatisierten Zugriff auf die erzeugten Report Files. Das ist beispielsweise dann der Fall wenn ein FPGA Design wiederholt implementiert werden muss um ein fehlerfreies Timing zu erhalten (auch bekannt als „Timing Closure“). In diesem Fall muss nach einer Implementierung die entsprechende Timing -Information automatisch aus dem Report File gelesen und die Implementierung gegebenenfalls mit einer anderen Strategie neu gestartet werden. Die Erzeugung solcher Tcl Skripte ist ebenfalls ein Schwerpunkt dieser Schulung.

Anwendbare Technologien

Alle von VIVADO unterstützten FPGA/SoC Technologien

Voraussetzungen

Grundlagen des VIVADO Design Flows

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Vivado „GUI“ vs. „Tcl-Scripting“

Überblick Tcl Scripting

- „Project“ vs. „Non-Project“-Mode
- Beispiele Project/Non-Project

Anpassung der Tcl Skripte

- Analyse der Report Files
- Definition verschiedener Synthese/Implementierungs-Läufe
- Automatischer Vergleich der erzielten Ergebnisse

Tcl – Konzept und Syntax

- Korrektes „Quoting“ von Sonderzeichen
- Umgang mit Listen

Unterprogramme

- Parameter-Übergabe
- Ergebnis-Rückgabe
- Fehlerbehandlung

Zeichenketten-Verarbeitung

- Sub-Operationen des Tcl ‚string‘-Kommandos
- String-Vergleich und -Zerlegung mit „Regulären Ausdrücken“

Datei-Verarbeitung

- Navigieren im Dateisystem
- Explizites Schreiben von Dateien
- Implizites Schreiben von Dateien
- Lesen von Textdateien
- Übungen

Navigieren in den Objekten des Design-Modells

- mit „hierarchischen Pfad-Angaben“
- mit „Regulären Ausdrücken“

Systematik der Vivado-Kommandos

- Analyse Vivado-Reports
- Zeichenketten Verarbeitung

Musterskripte für die automatische Implementierung

- Incremental Compile
- Multiple Runs & VIVADO Strategien
- Timing Closure mit Tcl Skript
- Übungen

Fortgeschrittene Tcl-Features

Workshop Designing with the XILINX Analog Mixed Signal Solution

Die Analog Mixed Signal Verarbeitung ist heute bereits mit FPGAs anwendbar!

Dieser Kurs führt in die Analog Mixed Signal Design Lösung mit XILINX FPGAs und die entsprechenden Werkzeuge und Techniken ein.

Der Kurs richtet sich an Hardware-Ingenieure der Analog und Digitaltechnik, verschiedene AMS Techniken nutzen zu können, wobei insbesondere der XADC Core genutzt werden wird.

Der komplette Front-to-Back-Design-Flow ist abgedeckt, Testsignalergzeugung, Einbinden der digital verarbeitenden Einheiten und insbesondere die Auswertung mittels dem XILINX Analog-to-Digital-Converter (XADC) Block.

Unter Verwendung eines XILINX Evaluierungs-Boards mit FMC-Add-on-Karte werden mit begleitenden Übungen die

verschiedenen Möglichkeiten praktisch angewandt, von der XADC Simulation eines analogen Eingangssignals bis zur digitalen Auswertung im FPGA Design, wie z.B. der FFT Analyse.

Anwendbare Technologien

XILINX 7-Series FPGAs

Voraussetzungen

Kenntnis der FPGA Technologie

Kenntnis der XILINX Design Tool Flows

Grundkenntnisse VHDL

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- AMS Overview
- ADC Theory
- DAC Theory
- XADC Architecture
- AMS Design Flow
- Lab View Umgebung
- XADC Debug and Monitor
- HDL Design Flow
- Embedded Design Flow
- ISE/Vivado Design Flow
- Course Summary

Übungen:

- **Generieren eines Digital/Analogen Quellsignals**
Erstellung von HDL Simulationsmodellen und HDL Simulation

- **Evaluieren des XADC Block**
Verwenden Sie das AMS TRD, um die XADC Leistung zu bewerten. Konfiguration der XADC Einstellungen und Verifikation der abgetasteten Daten mit der Lab View GUI-Umgebung.
- **AMS HDL State Machine Controller und Simulation**
Erstellung eines XADS cores unter Verwendung von Tool Wizard, RTL Projekt Managment mit vollständigem Design Flow, von Simulation und Implementierung.
- **Embedded Design Flow**
Mit Unterstützung eines MicroBlaze Controller lassen sich insbesondere Analysen der Signale effizient durchführen. In diesem Lab wird der XADC Core mit dem Prozessor konfiguriert, so dass unter der Programmiersprache C die Testanwendungen über den XADC Core laufen.

Workshop

SysML-Einführung für Systemingenieure

Wir zeigen, wie Systemingenieure mit Modellen nicht nur die Anforderungen an ein System aufnehmen, sondern auch dessen Struktur und Verhalten beschreiben. Sie erhalten eine Einführung, wie Sie anhand von Modellen die Schnittstellen zu anderen Systemen darstellen, funktionale Sicherheit analysieren und dabei auch Faktoren wie Entwicklungs- und Betriebskosten, Wartung, Systemperformance, Test, Produktion und Wiederverwendung berücksichtigen können.

Dabei dient die Systems Modeling Language (SysML) als Grundlage für eine interdisziplinäre Systembeschreibung und eine ganzheitliche, modellbasierte Entwicklungsme-

thodik. Anforderungen, Analyse, Design, Simulationen und Tests werden mit Diagrammen und Modellelementen dieser Notation beschrieben und im Rahmen eines Beispiels erklärt.

Voraussetzungen

Grundlegende Kenntnisse des Anforderungsmanagements und der UML

Dauer und Kosten

1 Tag, € 750,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Überblick Prozess- und Projektziele

Systeme, Embedded Systeme, Systems Engineering

MBSE: Modellbasiertes Systems Engineering mit SysML

- Funktionale Anforderungen
- Systemkontext
- Funktionales / logisches Modell
- Konkretes Modell
- Testen und Traceability
- Nicht-funktionale Anforderungen

Ausbildungsschwerpunkt „Embedded“

Die in diesem Bereich zusammengefassten Workshops beinhalten alle Informationen die Sie benötigen um Embedded Systeme mit XILINX FPGAs bzw. SoCs/MPSoCs entwickeln zu können. Der Entwicklungsvorgang eingebetteter Systeme ist in Bereiche Softwaredesign und Hardwaredesign gegliedert. Das Angebot rund um den Hardware Entwurf beinhaltet die Entwicklung der Embedded Plattform basierend auf MicroBlaze, ZYNQ-7000 SoC, ZYNQ UltraScale+ MPSoC oder Versal ACAP Architekturen.


Der softwarespezifische Teil beinhaltet Schulungsthemen rund um C/C++ sowie Linux und Treiberentwicklung unter Linux. Hier kooperieren wir mit auf diesem Gebiet international anerkannten Experten.

Manche der in diesem Abschnitt zusammengefassten Schulungen sind für Neueinsteiger im Bereich Embedded, andere wiederum sind mehr für erfahrene embedded Entwickler geeignet. Selbstverständlich können Sie alle hier vorhandenen Themen in beliebiger Kombination auch in einer In-House Schulung kombinieren. Informieren Sie sich auch über unsere berufsbegleitende Ausbildung zum Experten im Bereich FPGA Embedded Design.

Sie können sich jederzeit von unseren erfahrenen Trainern und Referenten im Vorfeld beraten lassen.

Neben den bereits bekannten und bewährten Schulungen bieten wir „Vitis HLS“ als neuen Workshop an.

Embedded	Seite	Dauer (Tage)	Kosten
ZYNQ UltraScale + MPSoC System Architecture	54	2	€ 1.500 o. 18 TCs
Comp. ZYNQ UltraScale+ MPSoC HW Designers	55	3	€ 2.100 o. 27 TCs
Adv. ZYNQ UltraScale+ MPSoC HW Designers	56	3	€ 2.100 o. 27 TCs
Comp. ZYNQ UltraScale+ MPSoC SW Designers	57	3	€ 2.100 o. 27 TCs
Professional ZYNQ UltraScale + MPSoC	58	5	€ 3.100 o. 45 TCs
ZYNQ-7000 SoC System Architecture	59	2	€ 1.500 o. 18 TCs
Compact ZYNQ-7000 SoC for HW Designers	60	3	€ 2.100 o. 27 TCs
Compact ZYNQ-7000 SoC for SW Designers	61	3	€ 2.100 o. 27 TCs
Professional ZYNQ-7000 SoC	62	5	€ 3.100 o. 45 TCs
Expert ZYNQ-7000 SoC	63	5	€ 3.100 o. 45 TCs
Compact VITIS for Software Designers	64	3	€ 2.100 o. 27 TCs
Compact VITIS for Acceleration	65	3	€ 2.100 o. 27 TCs

Embedded	Seite	Dauer (Tage)	Kosten
VITIS HLS 	66	3	€ 2.100 o. 27 TCs
Compact VITIS AI	67	3	€ 2.100 o. 27 TCs
Professional VITIS	68	5	€ 3.100 o. 45 TCs
Versal ACAP System Architecture	69	2	€ 1.500 o. 18 TCs
Compact Versal ACAP for HW Designer	70	3	€ 2.100 o. 27 TCs
Compact Versal ACAP for SW Designer	71	3	€ 2.100 o. 27 TCs
Professional Versal ACAP	72	5	€ 3.100 o. 45 TCs
Advanced Versal AI Engine	73	3	€ 2.100 o. 27 TCs
Expert Versal AI Engine	74	5	€ 3.100 o. 45 TCs
AXI Interface Technology	75	2	€ 1.500 o. 18 TCs
High-Speed Ethernet – Hands-On System Dev.	76	3	€ 2.100 o. 27 TCs
PCI Express – Hands-On System Development	77	3	€ 2.100 o. 27 TCs
Designing with RF Data Converters	78	3	€ 2.100 o. 27 TCs
Professional RFSoc	79	5	€ 3.100 o. 45 TCs
Professional MicroBlaze System Design	82	5	€ 3.100 o. 45 TCs
Dev. Multimedia Solutions VCU and GStreamer	83	2	€ 1.500 o. 18 TCs
Compact Embedded Linux	84	3	€ 2.100 o. 27 TCs
Embedded Linux Development Yocto Project	85	2	€ 1.500 o. 18 TCs
Embedded Linux Treiberentwicklung	86	3	€ 2.100 o. 27 TCs
Embedded Design with Petalinux Tools	87	2	€ 1.500 o. 18 TCs
Compact Python for Embedded	88	3	€ 2.100 o. 27 TCs
Professional Python for Embedded	89	5	€ 3.100 o. 45 TCs
Migrate to Real-Time OS: Hands-On Syst. Dev.	90	2	€ 1.500 o. 18 TCs
Real-Time Control Sys. Dev. using RTOS	91	2	€ 1.500 o. 18 TCs
Essentials of Microprocessors	93	1	€ 750 o. 9 TCs
RISC-V Architecture and FPGA implementation	94	2	€ 1.500 o. 18 TCs
RISC-V Core Verification & Compliance Testing	95	2	€ 1.500 o. 18 TCs



Workshop ZYNQ UltraScale+ MPSoC System Architecture

In diesem 2-tägigen Workshop wird auf die Systemarchitektur des XILINX ZYNQ UltraScale+ MPSoC Bausteins eingegangen. Nach einem Gesamtüberblick wird auf die im Processing System (PS) dediziert implementierten Prozessor-Einheiten eingegangen. Besonders das Power-Management mit der Platform Management Unit (PMU) im PS ist ein sehr hilfreiches Feature der ZYNQ UltraScale+ MPSoC Architektur. Des weiteren bietet die PS Architektur auch ein Datenbus-Management mit System-Kohärenz auf Hardwareebene für PS und PL Zugriffe. Anschließend werden system-spezifische Merkmale, wie System Protection und Takt- und Resetstrukturen besprochen. Nach einem Abschnitt über den im PS implementierten DDRAM Controller wird der Boot- und Konfigurationsprozess des ZU+ MPSoC Bausteins im Detail besprochen. Während der einzelnen Kapitel werden verschie-

dene Übungen, die eigenständig von den Teilnehmern durchgeführt werden, zur Verfügung gestellt.

Anwendbare Technologien

XILINX ZYNQ UltraScale+ MPSoC & RFSoc

Voraussetzungen

Erfahrungen digitaler Systemarchitekturen
Grundlagenkenntnisse Mikroprozessorarchitektur
Grundlagenkenntnisse Programmiersprache C

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Day 1:

- ZYNQ UltraScale+ MPSoC Overview
- ZYNQ UltraScale+ MPSoC Application Processing Unit
- ZYNQ UltraScale+ MPSoC Real-Time Processing Unit
- ZYNQ UltraScale+ MPSoC HW-SW Virtualization
- ZYNQ UltraScale+ MPSoC Power Management
- ZYNQ UltraScale+ MPSoC System Coherency
- ZYNQ UltraScale+ MPSoC System Protection

Day 2:

- ZYNQ UltraScale+ MPSoC Clocks and Resets
- ZYNQ UltraScale+ MPSoC DDR and QoS
- ZYNQ UltraScale+ MPSoC Boot and Configuration
- ZYNQ UltraScale+ MPSoC Ecosystem Support

Themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung

ZYNQ® UltraSCALE+ Workshop Compact ZYNQ UltraScale+ MPSoC for HW Designers

Dieser Kurs vermittelt sowohl die Tool-, als auch die Architektur-spezifischen Aspekte, die für die Entwicklung mit dem XILINX ZYNQ UltraScale+ MPSoC Baustein notwendig sind. Zu Beginn wird speziell auf den Embedded Design Flow eingegangen. Der Fokus in diesem Kurs liegt auf der Embedded Hardwareentwicklung mit dem XILINX VIVADO Tool, wobei auch auf die Softwareentwicklung mit der XILINX SDK eingegangen wird. Anschließend wird die Gesamtarchitektur des ZYNQ UltraScale+ MPSoC Bausteins besprochen. Für die Anbindung von AXI-basierten IPs in der Programmierbaren Logik (PL) an das Processing System (PS) ist es notwendig das AXI Protokoll und auch die Interrupt-Strukturen genau zu verstehen. Der letzte Abschnitt dieses Kurses besteht aus der Erstellung und der Verifikation eigen

erstellter IP Cores mit AXI Schnittstellen.

Anwendbare Technologien

XILINX ZYNQ UltraScale+ MPSoC & RFSoc

Voraussetzungen

Verständnis digitaler Systemarchitekturen
VHDL und FPGA Kenntnisse sind von Vorteil
Grundlagen Programmiersprache C ist von Vorteil

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Embedded UltraFast Design Methodology
- Overview of Embedded Hardware Development
- Driving the IP Integrator Tool
- Driving the Vitis Tools
- ZYNQ UltraScale+ MPSoC Architecture Overview
- ZYNQ UltraScale+ MPSoC System Coherency
- ZYNQ UltraScale+ MPSoC System Protection
- Introduction to AXI
- ZYNQ UltraScale+ MPSoC Interrupts: Hardware Aspects
- Adding and Connecting AXI IP
- Using the Create and Import Wizard to Create a New AXI IP
- BFM Simulation

Themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung

ZYNQ[®] UltraSCALE+ **Workshop Advanced ZYNQ** **Ultrascale+ MPSoC for HW Designers**

Dieser Workshop vermittelt sowohl die Tool-, als auch die Architektur-spezifischen Aspekte, die für die Entwicklung mit der XILINX ZYNQ UltraScale+ MPSoC Architektur grundlegend sind. Der Schwerpunkt dieses Kurses liegt auf dem Verständnis und dem Arbeiten mit der embedded Hardware Architektur unter Nutzung dem XILINX VIVADO-Tools mit Verwendung des IP-Integrators, aber auch des Software Tools Vitis, in der die Hardware Inbetriebnahme die erforderliche Software Unterstützung bekommt. Die Gesamtarchitektur des ZYNQ UltraScale + MPSoC-Verarbeitungssystem (PS) wird ausführlich erläutert, um die Architektur des Prozessor Systems (PS) zu verstehen, das wiederum AXI-basierte Schnittstelle für die programmierbare Logik (PL) zur Verfügung stellt. Die APU beinhaltet die ARM Cortex-A53-Kerne, die RPU wiederum die Cortex-R5-Kerne und die PMU wiederum ein MicroBlaze-System. So wird es auch notwendig sein, Zugriffe in einem System aus gemeinsam genutzten Peripherien und gemeinsamem Speicher zu schützen und zu isolieren, wenn MPSoC-basierte Prozesse gleichzeitig

in Ausführung sind. Für diese Verbindung von AXI-basierten IPs in der Programmable Logic (PL) mit dem Processing System (PS) ist es wichtig, das AXI-Protokoll mit Funktionen wie Kohärenzmanagement oder virtuelles Management zu verstehen. Der letzte Teil dieses Kurses umfasst Konzepte des Boot-Managements, Plattformmanagements, Energieverwaltung und Interprozessor-Interrupts.

Anwendbare Technologien

XILINX ZYNQ UltraScale+ MPSoC & RFSoc

Voraussetzungen

Verständnis digitaler Systemarchitekturen
VHDL und FPGA Kenntnisse
Grundlagen Programmiersprache C

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Zynq US+ MPSoC APU Cortex-A53 Processor
- Zynq US+ MPSoC APU Architecture Extensions
- Zynq US+ MPSoC APU 64-Bit Architecture Features
- Zynq US+ MPSoC APU Exception Handling
- Zynq US+ MPSoC Cache Coherency
- Hypervisors Introduction
- Zynq UltraScale MPSoC Virtualization Hardware Support
- Zynq US+ MPSoC RPU Cortex-R5 Architecture
- Zynq US+ MPSoC RPU Clocking, Power, and Reset
- Zynq US+ MPSoC RPU TCM Architecture
- QEMU Introduction and Launching
- Zynq US+ MPSoC Boot and Configuration
- Zynq US+ MPSoC Boot Image
- FSBL Introduction
- Introduction to Video and Video Codecs
- Zynq US+ MPSoC Video Codec Unit
- Zynq US+ MPSoC System Memory Management Unit
- Zynq US+ MPSoC Peripheral and Memory Protection Units
- Zynq US+ MPSoC Memory Protection Unit
- Zynq US+ MPSoC Clocking and PS Resets
- AXI Introduction
- AXI Variations and Transactions
- Zynq US+ MPSoC PMU Introduction
- Zynq US+ MPSoC PMU Hardware Architecture
- Zynq US+ MPSoC PMU and the IPs

Themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung

Workshop Compact ZYNQ UltraScale+ MPSoC for SW Designers

Dieser 3-tägige Kurs dient dazu dem Softwareentwickler den bestmöglichen Einstieg in die Softwareentwicklung für die ZYNQ UltraScale+ MPSoC Familie zu ermöglichen. Die Processing System MPSoC Architektur wird erläutert und das Arbeiten mit verschiedenen Methoden, die für die Softwarephase im Embedded Designzyklus notwendig oder hilfreich sind mit den unterstützten Betriebssystemen Linux, FreeRTOS und BareMetal. Die Open-Source Linux Unterstützung, Unterstützung mehrerer Betriebssysteme, ob symmetrisch (SMP) oder asymmetrische (AMP), das OpenAMP Framework, FreeRTOS für die Unterstützung der Real-Time Prozessoren sowie die Mechanismen von Boot-Konfigurationen werden aufgezeigt und in Übungen erarbeitet. Das Debugging in der Simulation, unterstützt durch QEMU oder

aber auf der Hardware, ist wichtiger integraler Bestandteil. Mit sieben Prozessoren in der UltraScale+ MPSoC Architektur ist dazu das dynamische Power Management Software-programmierbar, dessen Methoden ebenso vermittelt werden.

Anwendbare Technologien

XILINX ZYNQ UltraScale+ MPSoC & RFSoc

Voraussetzungen

Verständnis von digitalen embedded Systemen
Grundlagenkenntnisse Programmiersprache C

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- ZYNQ UltraScale+ MPSoC Processing Units
- ARM TrustZone Technology
- ZYNQ UltraScale+ MPSoC HW-SW Virtualization
- The XILINX Vitis Tools
- Introduction to QEMU
- Bare-Metal Application Development and Debugging
- Linux Application Development and Debugging
- Deploying OpenAMP in a Heterogeneous System
- Symmetric Multi-Processing Linux
- PetaLinux and Yocto
- Open Source Library-PetaLinux Tools
- Understanding Device Drivers
- ZYNQ UltraScale+ MPSoC FreeRTOS
- ZYNQ UltraScale+ MPSoC Software Stack
- ZYNQ UltraScale+ MPSoC PMU Development and Debugging
- ZYNQ UltraScale+ MPSoC Power Management
- ZYNQ UltraScale+ MPSoC Boot and Configuration

Themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung



PowerWorkshop Professional ZYNQ UltraScale+ MPSoC

Dieser Power-Workshop “Professional Zynq UltraScale+ MPSoC” ist eine Kombination aus den beiden Kursen “Compact Zynq UltraScale+ MPSoC for Hardware Designers” und “Compact Zynq UltraScale+ MPSoC for Software Designers”.

Dieser vermittelt das notwendige und auch tiefgreifende Wissen, um den gesamten Embedded Design Zyklus für den Zynq UltraScale+ MPSoC zu verstehen und die dazu notwendigen Tools sicher anwenden zu können.

Die hierbei verwendeten Tools sind der IP Integrator (IPI) unter Vivado und die XILINX Unified Platform Vitis für die jeweilige Erstellung der embedded Hardware- und Software-schicht. Zudem wird auch auf verschiedene architekturenspezifische Eigenschaften des Zynq UltraScale+ MPSoCs eingegangen, u.a. auf die System-Kohärenz, System-Protection oder auch auf die vorhandenen AXI Schnittstellen zwischen dem Processing System (PS) und der Programmierbaren Logik (PL).

Der zweite Teil dieses Kurses besteht aus Software-spezifischen Inhalten. Hierbei werden die von XILINX zur Verfügung gestellten Softwaretreiber, Bibliotheken und weitere

Services erläutert, die dem Softwareentwickler mit der Softwareentwicklung zur Verfügung stehen. Die Open-Source Linux Unterstützung mit Yocto- oder PetaLinux-Build für die APU wird ebenso behandelt wie die FreeRTOS Unterstützung und das Multi-OS Management mit Datenaustausch zwischen den Systemen.

Im letzten Teil wird die Software-konfigurierbare Power Management Unit behandelt sowie der Boot- und Konfigurationsprozess mit First-/Second-Stage Boot, Bitstream loading, Multiboot, Secure-Boot und der Boot-Image Generierung.

Anwendbare Technologien

XILINX FPGAs, SoC, MPSoC & RFSoc

Voraussetzungen

Kenntnisse von digitalen embedded Systemen

Grundlagenkenntnisse Programmiersprache C

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Embedded UltraFast Design Methodology
- Overview of Embedded Hardware Development
- Driving the IP Integrator Tool
- Driving the SDK Tool
- Zynq UltraScale+ MPSoC Architecture
- Zynq UltraScale+ MPSoC System Coherency
- Zynq UltraScale+ MPSoC System Protection
- Zynq UltraScale+ MPSoC Interrupts
- Adding and Connecting AXI IP
- Creating User Peripherals based on AXI
- Overview of Embedded Software Development
- Standalone Software Platform Development
- ARM TrustZone Technology
- Zynq UltraScale+ MPSoC HWSW Virtualization
- Driving the Xilinx Vitis Tool
- Introduction to QEMU
- BareMetal Application Development and Debugging
- Linux Application Development and Debugging
- Deploying OpenAMP in a Heterogeneous System
- Symmetric MultiProcessing Linux
- PetaLinux and Yocto
- Open Source-Linux PetaLinux Tools
- Understanding Device Drivers
- Zynq UltraScale+ MPSoC FreeRTOS
- Zynq UltraScale+ MPSoC Software Stack
- Zynq UltraScale+ MPSoC PMU Development and Debugging
- Zynq UltraScale+ MPSoC Power Management
- Zynq UltraScale+ MPSoC Boot and Configuration

**Umfangreiche themenbegleitende Übungen
möglichen eine praxisgerechte Einarbeitung**

ZYNQ **Workshop ZYNQ-7000 SoC System Architecture**

Dieser Kurs vermittelt erfahrenen Systemarchitekten das Wissen um sehr effektiv mit der bewährten ZYNQ-7000 SoC Architektur zu arbeiten.

Ziel dieses 2-tägigen Kurses ist es dem Teilnehmer die Funktionen und Vorteile der ZYNQ Architektur näher zu bringen, damit Entscheidungen während eines Projektes mit der ZYNQ Architektur einfacher getroffen werden können. Die Architektur des ARM Cortex-A9 Prozessor-basierenden Processing Systems (PS) und die Integration der Programmable Logic (PL) werden in einer ausreichend tiefen Ebene besprochen, so dass der Systementwickler erfolgreich und effektiv Projekte mit dem ZYNQ-7000 Baustein durchführen kann.

Besprochen werden hier unter Anderem die verschiedenen Komponenten im Processing System, wie I/O Peripherien, Timer, Caches, sowie die DMA Controller, das Interrupt-handling und die Memory Controller. Der Schwerpunkt liegt hierbei auf der effektiven Verwendung des PS DDR Controllers aus der Sicht der PL, das Interfacing zwischen PS und PL, Designtechniken und die Vorteile der Implementierung von Funktionen im PS oder in der PL.

Um tiefere Informationen über die entsprechenden Entwicklungstool zu erhalten, bietet sich der PLC2 Power-Workshop „Professional ZYNQ-7000 SoC“, sowie die 3-Tages Varianten „Compact ZYNQ-7000 SoC for Hardware Designers“ und „Compact ZYNQ-7000 SoC for Software Designers“ an.

Anwendbare Technologien

XILINX ZYNQ-7000 SoCs

Voraussetzungen

Grundlagenkenntnisse von Embedded Controllern
Grundlagenkenntnisse FPGA Technologie
Grundlagenkenntnisse Vitis Unified Platform
Grundlagenkenntnisse Programmiersprachen VHDL und C

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- ZYNQ-7000 SoC Overview
- Inside the Application Processor Unit (APU)
- **Lab 1:** Building a ZYNQ-7000 SoC Platform
- Processor Input/Output Peripherals
- Introduction to AXI
- ZYNQ-7000 SoC PS-PL Interface
- **Lab 2:** Integrating Programmable Logic on the ZYNQ-7000 SoCs
- ZYNQ-7000 SoC Booting
- **Lab 3:** Using DMA on the ZYNQ-7000 SoC
- ZYNQ-7000 SoC Memory Resources
- Meeting Performance Goals
- **Lab 4:** SPM – Impact of Port Selection on System
- ZYNQ-7000 SoC Hardware Design
- ZYNQ-7000 SoC Software Design
- Debugging the ZYNQ-7000 SoC
- **Lab 5:** Debugging on the ZYNQ-7000 SoC
- ZYNQ-7000 SoC Tools and Reference Designs
- **Lab 6:** Running and Debugging a Linux Application on the ZYNQ-7000 SoC



Workshop Compact ZYNQ-7000 SoC for HW Designers

Der Schwerpunkt des Workshops „Compact ZYNQ-7000 SoC for HW Designers“ liegt in der Erstellung von Hardware Plattformen für die bewährte ZYNQ-7000 SoC Technologie, sowie in dem Umgang mit der Xilinx Vitis Unified Development Platform.

Zu Beginn des Workshops lernt der Systementwickler die grundlegenden Architekturelemente der ZYNQ-7000 Technologie kennen. Zum einen wird auf das Prozessorsystem (PS) eingegangen. Hier werden die einzelnen Komponenten im Detail besprochen und die Konfigurationsmöglichkeiten des PS erläutert. Zum Anderen wird die Programmierbare Logik (PL) und das AXI Interface zu dem Prozessorsystem genauer erläutert. Ein weiterer Themenschwerpunkt widmet sich dem Erstellen und Simulieren von maßgeschneiderten AXI4-basierenden Peripherien. Zusätzlich wird das Interrupt Handling, sowie das Hinzufügen und Downloaden der Software diskutiert. Das erlernte Wissen wird durch eigenständig durch-

geführte Übungen angewendet.

Um den Fokus weiter auf die Software Entwicklung mit Vitis zu richten bietet sich der Workshop „Compact ZYNQ-7000 SoC for SW Designers“ als Aufbaukurs an.

Anwendbare Technologien

XILINX ZYNQ-7000 SoC

Voraussetzungen

VHDL und FPGA Kenntnisse

Grundlagen der Programmiersprache C

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen.

Agenda

- Embedded Design Overview
- IP Integrator and the PS Configuration Wizard
- Software Development Using Vitis
- Introduction to AXI
- Interrupts
- Adding Hardware to an Embedded System
- Cortex-A9 Processor Basics
- Designing a Custom AXI Peripheral
- Using the Create and Package IP Wizard to Build a Custom AXI Peripheral
- Bus Functional Model Simulation



Workshop Compact ZYNQ-7000 SoC for SW Designers

Der Workshop „Compact ZYNQ-7000 SoC for SW Designers“ behandelt im Wesentlichen die softwarebasierenden Schwerpunkte der ZYNQ-7000 System on a Chip (SoC) Architektur. Im Mittelpunkt steht hierbei die Xilinx Vitis Softwareumgebung. Die Teilnahme an diesem 3-tägigen Kurs bietet sich sowohl für den Software- als auch für den Hardwareentwickler an.

Dieser Workshop bietet dem Embedded Software Designer die erforderlichen Kenntnisse, um eine softwarebasierende Embedded Applikation erfolgreich zu gestalten. Der Teilnehmer erlernt den Umgang mit dem XILINX Vitis Entwicklungstool und Konzepte die notwendig für die Softwarephase in einem Designzyklus sind. Im Fokus hierbei steht das Prozessorsystem PS. Die Beschreibung des Prozessors und der Peripherien ist ebenso ein Themenschwerpunkt wie das Schreiben von Softwarecode unter Berücksichtigung des Adressmanagements und des Interrupthandlings. Anschließend wird das Downloaden und Booten der Softwareplattform und das Debuggen des

C-Codes diskutiert. Um das Wissen weiter zu vertiefen bietet sich der 5-tägige PowerWorkshop „Expert ZYNQ-7000 SoC“ an.

Eine Hardware Plattform wird mit dem IP-Integrator der Vivado Tool Suite erstellt, was nicht im Fokus dieses Workshops steht, sondern die Projektierung von embedded Software für die Cortex-A9 basierende Zielplattform der ZYNQ-7000 SoC Familie.

Anwendbare Technologien

XILINX ZYNQ-7000 SoC Familie

Voraussetzungen

Grundkenntnisse Mikrocontroller

Grundlagen der Programmiersprache C

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen.

Agenda

- Course Agenda and Introduction
- Overview of Embedded Software Development
- Embedded UltraFast Design Methodology
- Zynq-7000 SoC Architecture Overview
- Driving the Vitis Software Development Tool
- System Debugger
- Standalone Software Platform Development and Coding Support
- FAT File System
- Using Linker Scripts
- Migrating from SDK to the Vitis Platform
- Introduction to Interrupts, Writing S/W Interrupts
- Operating Systems Introduction and Concepts
- Linux A High-Level Introduction
- Linux Software Application Development Overview
- Building a Linux Application in the Vitis IDE
- Booting Overview
- Software Profiling Overview
- Understanding Device Drivers
- Custom Device Drivers

Umfangreiche themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung



PowerWorkshop Professional ZYNQ-7000 SoC

Die ZYNQ-7000 SoC Technologie von XILINX hat einen sehr großen Anklang in der Welt der Embedded Anwendungen gefunden.

Dieser 5-Tages PowerWorkshop „Professional ZYNQ-7000 SoC“ spricht sowohl Hardware- und Softwareentwickler an, um erfolgreich Embedded-Projekte mit dem ZYNQ-7000 Baustein durchzuführen.

Neben einer umfangreichen Architektur-Beschreibung besteht dieser Kurs unter Anderem aus dem Umgang mit den XILINX Entwicklungstools speziell für die ZYNQ-7000 Technologie. Außerdem wird im Detail auf das ARM-zentrische Prozessorsystem PS und die AXI-basierenden Interconnect Strukturen zur Programmierbaren Logik PL eingegangen. Das Erstellen eigener AXI4-basierender Peripherien ist ebenso Bestandteil dieser Schulung. Der zweite Teil dieser 5 Tage behandelt die Software Aspekte des ZYNQ-7000 Bausteins, wie z.B. das Projektie-

ren der Software mit der Vitis Unified Development Platform. Weitere Themen sind u.a. die Interrupt-Verwaltung, das Adressmanagement, das Erstellen von Softwaretreibern, das Debugging und Profiling und der Boot Prozess des ZYNQ-7000 Systems.

Wie bei allen Workshops der PowerWorkshop-Reihe werden die Themeninhalte durch vielfältige Übungen vertieft.

Anwendbare Technologien

XILINX ZYNQ-7000 SoC

Voraussetzungen

VHDL und FPGA Kenntnisse

Grundlagen der Programmiersprache C

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen.

Agenda

- Embedded Design Overview
- IP Integrator and the PS Configuration Wizard
- Software Development Using Vitis
- Introduction to AXI
- Interrupts
- Adding Hardware to an Embedded System
- Cortex-A9 Processor Basics
- Designing a Custom Peripheral
- Using the Create and Package IP Wizard to Build a Custom AXI Peripheral
- Bus Functional Model Simulation
- Software Platform Development
- Software Development Using Vitis
- Writing Code in the XILINX Environment
- Address Management
- Software Interrupts
- Software Platform Download and Boot
- Application Debugging
- Application Profiling
- Writing a Custom Device Driver
- Advanced Services and Operating Systems
- Project Management with the XILINX Design Tools



PowerWorkshop Expert ZYNQ-7000 SoC

Der PowerWorkshop „Expert ZYNQ-7000 SoC“ ist als Erweiterung des PowerWorkshops „Professional ZYNQ-7000 SoC“ gedacht.

Er bietet dem Teilnehmer die Möglichkeit das Wissen über die neue ZYNQ-7000 SoC Architektur zu vertiefen und weiter auszubauen. Insbesondere der methodische Umgang mit den Xilinx Design-Suites von Vivado und Vitis spielen hier eine wichtige Rolle. Zusätzlich wird auf die verschiedenen Peripherien im Prozessorsystem genauer eingegangen. Desweiteren wird der Einsatz der Prozessoren im ZYNQ-7000 Baustein im Detail besprochen. Ein weiterer thematischer Schwerpunkt ist die Konfiguration eines DMA Controllers.

In dieser 5-tägigen Variante werden die Themeninhalte durch vielfältige Übungen intensiviert.

Das erlernte Wissen wird durch zahlreiche eigenständig

durchgeführte Übungen vertieft.

Es werden sowohl embedded Hardware Themen im Vivado Tool Flow als auch embedded Software Themen behandelt, in welchen Teilnehmer bereits Kenntnisse vorweisen sollten.

Anwendbare Technologien

XILINX ZYNQ-7000 SoC

Voraussetzungen

Grundlagenkenntnisse ZYNQ-7000 SoC Architektur
Grundlegende VHDL und C Kenntnisse

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen.

Agenda

- Embedded Systems Development Review
 - ZYNQ-7000 SoC Processing System Overview
 - Debugging Using the ChipScope Pro Analyzer
 - JTAG to AXI Master Debug IP Transactions
 - Block RAM Memory Controllers
 - External Memory Controllers for Static Memory
 - Memory Controllers for Dynamic RAM
 - Interrupts
 - AXI Streaming Interface
 - System Data Movement: Low Latency and High Bandwidth
 - Advanced Processor and Peripheral Interface Options
 - Advanced Processor Configurations
 - Software Boot and PL Configuration
 - HDL System Simulation with an Embedded Processor
 - Advanced Boot Methodology on the ZYNQ-7000 SoC
 - ZYNQ-7000 Boot Details
 - Creating a Boot Image
 - Advanced Cortex-A9 Processor Services
 - Advanced DMA Controller Configuration on the ZYNQ-7000 SoC
 - High-Speed Peripheral Configuration on the ZYNQ-7000 SoC
 - Low-Speed Peripherals on the ZYNQ-7000 SoC
 - Ethernet Support using LwIP Stack
- + zahlreiche Übungen

XILINX VITIS™ **Workshop** Compact Vitis for Software Designer

Dieser Kurs vermittelt die Embedded Software Entwicklung in der Xilinx Vitis Unified Software Plattform. Es wird die grundlegende Projekterstellung für die Zynq 7000 SoC und Zynq UltraScale+ MPSoC erarbeitet, wodurch der Kurs auch für Hardware Entwickler sinnvoll ist. Das Angebot richtet sich an Entwickler, die auf bereitgestellten FPGA Plattformen Anwendungen erstellen wollen. Die Teilnehmer lernen, die Konzepte der Vitis Umgebung im Projektzyklus zu verwenden. Beispielhaft wird auf den leistungsfähigen Prozessorsystemen der Zynq Familien gearbeitet, die mit ihren Peripherie-Elementen und deren Schnittstellen vorgestellt werden. Die Grundlagen der Software Entwicklung auf den hardwarenahen Bibliotheken dieser Bausteine werden verdeutlicht. Ein Einblick in Linux-artige Betriebssysteme wird gegeben, damit die Teilnehmer Linux Applikationen in der Vitis IDE erstellen lernen. Die in Vitis vorhandenen Möglichkeiten zum Debugging und Profiling werden verwendet, um diese Projekte auf Hardware zu betreiben.

Der Kurs behandelt die Themen:

- Einführung in die Vitis Software Development Umgebung
- Portierung von Vivado SDK auf die Vitis Plattform
- Anpassung der Board Support Packages (BSPs) auf Basis der Xilinx Standalone library
- Software Entwicklung für ARM® Cortex™-A9 / A53 Prozessoren

- Einsicht in Memory File Systeme mit der FFS library
- Effektiver Einsatz von Gerätetreibern für die Xilinx Prozessoren
- Debugging und Integration von eigenem Code
- Bewährte Techniken für solide Design Entscheidungen
- Praktische Übungen auf dem ZCU104 Board oder mit QEMU

Anwendbare Technologien

Architekturen: Zynq-7000 SoC (Cortex-A9 processor), Zynq UltraScale+ MPSoC (Cortex-A53 und Cortex-R5 processor), ZCU104 board

Voraussetzungen

Verständnis der Konzepte in Mikrocontroller Systemen, Grundlegende Kenntnisse in C/C++, generelle Debugging Techniken

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlicher Schulungsunterlagen sowie Pausengetränke und Mittagessen

Agenda

- | | |
|---|---|
| <ul style="list-style-type: none"> ■ Einführung in die Vitis Unified Software Plattform ■ Überblick über Embedded Software Development ■ Embedded UltraFast Design Methodology ■ Zynq UltraScale+ MPSoC Architektur Übersicht ■ Bedienung der Vitis Software Development Werkzeuge ■ System Debugger ■ Standalone Software Plattform Development und Coding Support ■ FAT File System ■ Nutzung von Linker Scripts | <ul style="list-style-type: none"> ■ Portierung aus Vivado SDK auf die Vitis Plattform ■ Einführung in Interrupts ■ Operating Systems: Einführung und Konzepte ■ Linux: Ein Überblick ■ Linux Software Anwendungsentwicklung ■ Benutzung der PetaLinux Tools ■ Erstellen einer Linux Anwendung mit der Vitis IDE ■ Überblick: Booting ■ Einführung in das Software Profiling ■ Einblicke in Gerätetreiber ■ Eigene Gerätetreiber |
|---|---|

XILINX VITIS™ Workshop Compact Vitis for Acceleration

Durch die sequenzielle Funktion von CPUs in traditionellen Embedded Systemen entstehende Engpässe können mit Parallelisierung und optimiertem Datenfluss, z.B. durch FPGAs, extrem gesteigert werden. Xilinx bietet für solche heterogenen Systeme mit CPU und FPGA mit der Vitis Unified Software Plattform ein Werkzeug, das Softwarefunktionen in spezifische Hardwarebeschleuniger

(Kerne) umsetzt. Diese Werkzeuge erlauben, solche Kerne auf höherer Code Ebene zu beschreiben und durchgängig bis auf das Target FPGA abzubilden. Es wird gezeigt, wie diese Kerne z.B. mittels OpenCL™ API in die Applikation eingebunden sind und synchronisiert werden, um die bestmögliche Leistung zu erreichen. Dieser Kurs vermittelt, wie Vitis in der Entwicklung benutzt wird, um neue oder bestehende C/C++ und RTL Applikationen für Datacenter (DC) und Edge Lösungen zu erstellen. Die Teilnehmer erlernen, solche Designs auf z.B. den XILINX ALVEO Accelerator Boards lauffähig zu generieren, auch mit Hilfe von Debugging und Profiling Methoden, die Vitis bietet.

Der Kurs bietet:

- Erstellen von Software Anwendungen mit dem OpenCL™ API für Hardware Kernels auf Alveo Accelerator Karten
- Ausführungs- und Datenfluss-Steuerung von OpenCL™ Anwendungen auf Embedded Systemen
- Einführung in den Vitis Toolflow auf Benutzeroberfläche

und Skript-basiert für DC und Embedded Anwendungen

- Verständnis des Vitis Platform Execution Model und XRT
- Erarbeitung der Kernel Entwicklung mit C/C++ und RTL
- Anwendung des Vitis Analyzer Tools auf Projektdaten
- Erklären von Optimierungsstrategien für Accelerator Designs

Anwendbare Technologien

Architektur: Xilinx Alveo Accelerator Karten, SoCs und ACAPs

Voraussetzungen

Grundkenntnisse der Xilinx FPGA, Architektur, Kenntnisse im Programmieren mit C/C++, Einsicht in den Software Entwicklungsprozess

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlicher Schulungsunterlagen sowie Pausengetränke und Mittagessen

Agenda

- Einführung in die Vitis Unified Software Plattform
- Vitis IDE Tools
- Vitis Command Line Flow
- Einführung in Methoden der Hardwarebeschleunigung
- Rechenlast-Auslagerung und Pfad-Optimierung
- Vorstellung der Alveo Data Center Accelerator Karten
- Einführung zur Nimble Cloud
- Vitis Execution Model und XRT
- Einführung in das OpenCL Framework
- Xilinx Techniken zur OpenCL Synchronisierung
- Arbeiten mit mehrdimensionalen Daten (NDRanges)
- Profiling und Debugging
- Einführung in C/C++ basierte Kernel
- Wiederverwendung bestehender IP als Beschleuniger mit dem RTL Kernel Wizard
- Methoden der Optimierung
- Kernel Optimierung auf der C/C++ Ebene
- Optimierung mit dem zentralen Rechner
- Optimierung der Gesamtleistung eines Designs
- Vitis Accelerated Libraries, BLAS, Fintech and OpenCV
- Optimierung der Speicher Transfers
- Datenflusskontrolle mit XDMA und QDMA
- Anwendung des Vitis Analyzer Tools


Workshop Vitis HLS


Seit der Erfindung der FPGAs hat sich die Entwicklungsmethodik von der Eingabe mit Stromlaufplan bis hin zur Modellierung eines Systems auf RTL Ebene stetig verändert. Allerdings erfordern diese Methoden zusätzlich zur Beschreibung der gewünschten Funktionalität auch die genaue Definition der im FPGA abzubildenden Strukturen wie z.B. Pipelining, Latenz, Datendurchsatz oder Fläche. Die HLS Methodik ermöglicht es, C/C++-basierte Algorithmen vom Prozessorsystem auf Hardware (programmierbare Logik) auszulagern und in der abstrakten Systemmodellierung die Code-generierte Architektur weiter zu optimieren, und das ohne Kenntnis von Hardware Programmiersprachen. Die Verwendung der ‚C‘-basierenden Programmiersprachen wie ‚C‘, ‚C++‘ und ‚SystemC‘ automatisiert die Implementierung und die Optimierung des FPGAs durch die Umsetzung der ‚C‘-basierenden Beschreibung auf die RTL Ebene. Die signifikanten Vorteile der neuen Methodik bei der Verifikation und Implementierung sind offensichtlich.

Z.B. kann ein in ‚C‘, ‚C++‘ oder ‚SystemC‘ beschriebener Algorithmus wahlweise auf Geschwindigkeit, Latenz, Fläche usw. automatisch umgesetzt werden und erlaubt somit einen einfachen Vergleich der Resultate. Die weit verbreitete Verifikation auf der ‚C‘ Ebene erlaubt zusätzlich die frühe Erkennung der Designfehler. Beide Vorteile können daher die Entwicklungszeit der FPGA Projekte erheblich verkürzen und die Qualität steigern.

Anwendbare Technologien

XILINX FPGA, SoCs, MPSoCs, RFSocS und ACAPs

Voraussetzungen

Grundkenntnisse in C Programmierung
Grundkenntnisse der FPGA Architektur

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen.

Agenda

Introduction to High-Level Synthesis

- Language Support
- Validation Flow

Using the Vitis HLS Tool

- Project Creation in the Vitis HLS Tool
- C Validation to IP Creation
- Vitis HLS Tool Directory Structure

IO Interfaces

- Block-Level I/O Protocols
- Port-Level I/O Protocols
- Creating AXI Interfaces

Pipelining for Performance

- HLS UltraFast Design Methodology
- Pipelining
- Dataflow Optimization

Optimizing Structures for Performance

- Arrays in HLS
- Array Optimizations

Reducing Latency

- Improving Latency
- Improving Area

- Controlling the Resources Used
- Controlling the Structure of the Design
- Importance of Arbitrary Precision Types

Introduction to the HLS Design Flow

- RTL vs C-Based Design
- Traditional RTL vs Vitis HLS Design Flow

HLS vs. Vitis Acceleration Flow

- Kernel Acceleration with Vitis
- What Users Need to Know to Be Highly Successful

Vitis HLS Tool: C Coding Rules and Tips

Übungen

- Introduction to the Vitis HLS Tool Flow
- Introduction to the Vitis HLS Tool CLI Flow
- Interface Synthesis
- Improving Performance
- Implementing Arrays as RTL Interfaces
- Improving Area and Resource Utilization
- HLS Flow - System Generation



Workshop Compact Vitis AI

In diesem Kurs wird das Vitis AI Development Toolkit für die AI-Inference auf Xilinx Hardware-Plattformen in Verbindung mit DNN-Algorithmen, Modelltraining, zugehörigen Frameworks für die Entwicklung und Bereitstellung auf Alveo™-Karten, Zynq SoCs und Zynq® UltraScale+™ MP-SoCs vorgestellt. Die Grundlagen von maschinellem Lernen (ML) und die Herausforderungen beim Training in neuronalen Netzen werden als allgemeine Einführung vorgestellt. Mainstream-Frameworks wie Caffe und Tensorflow werden in der Vitis Toolchain gezeigt und Modelle für verschiedene Deep-Learning-Aufgaben werden diskutiert. Anschließend wird ein Einblick in die voroptimierte Modellbibliothek, welche Modelle bietet, die problemlos auf Xilinx-Geräten bereitgestellt werden können, gegeben. Die Konzepte des Vitis AI-Entwicklungskits werden mit den framework-spezifischen Tools erklärt, darunter zum Beispiel „pruning“ und „optimization“ der trainierten Modelle. Aus den Ergebnissen generiert der AI Compiler Code, der dann auf einer bestimmten FPGA-Fabric-Mikroarchitektur, der DPU, ausgeführt werden kann. Es wird verdeutlicht, wie die Tools des Vitis AI-Entwicklungskits zur Analyse der Modelleistung und zum Debuggen verwendet werden können. Um ein vollständiges Deployment zu realisieren, werden in den letzten Kapiteln Vitis AI-Bibliotheken und -APIs vorgestellt und es wird gezeigt, wie sie für eine optimierte Inference in die DPU integriert werden können.

Der Schwerpunkt dieses Kurses liegt auf:

- Veranschaulichung des Vitis AI-Tool flows
- Nutzung der architektonischen Features der Deep Learning Processor Unit (DPU)
- Optimieren eines Modells mit dem AI-Quantizer und dem AI-Compiler
- Verwendung der Vitis AI-Bibliothek zur Optimierung der pre- und post-processing Funktionen
- Erstellen einer benutzerdefinierten Plattform und Anwendung
- Bereitstellen eines Designs

Anwendbare Technologien

Architektur: Xilinx Alveo accelerator cards, Xilinx, SoCs & MPSoCs und ACAPs

Voraussetzungen

Grundkenntnisse in maschinellen Lernkonzepten, Basiswissen in den Programmiersprachen C / C ++ / Python, Softwareentwicklungsabläufe

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlicher Schulungsunterlagen sowie Pausengetränke und Mittagessen

Agenda

- Einführung in die Vitis AI Development Environment
- Übersicht über ML-Konzepte
- Frameworks, die von dem Vitis AI Development Environment unterstützt werden
- Einrichten des Vitis AI Development Environment
- AI Optimizer
- AI Quantizer und AI Compiler
- AI Profiler und AI Debugger
- Einführung in die Deep Learning Processor Unit (DPU)
- DPUs für verschiedene Xilinx Zielarchitekturen
- Vitis AI Library für effiziente AI Inferenz
- Erstellen einer benutzerdefinierten Hardwareplattform mit Vivado
- Erzeugen eines DPU Kernels in Vitis (Edge)
- Verwendung von Vitis zu Generieren einer Embedded Acceleration Plattform (Edge)
- Erstellen einer eigenen Applikation auf der Plattform (Edge)



Workshop Professional Vitis

Durch die sequenzielle Funktion der CPUs in Embedded Systemen entstehende Engpässe können mit der inhärenten Parallelität und dem anpassbaren Datenfluss von FPGAs extrem gesteigert werden. Xilinx bietet für solche heterogenen Systeme mit CPU und FPGA mit der Vitis Unified Software Platform ein Werkzeug, in dem sowohl Embedded Software Projekte als auch abstrakte Beschreibungen von Beschleunigern (Kernels) für ein Hardware Projekt gemeinsam verwaltet werden. Dieser Kurs ermöglicht Software Entwicklern, mit der Vitis Plattform Projekte für Xilinx Zynq SoC und MPSoC Familien zu erstellen. Es werden Kenntnisse in der Softwareentwicklung vermittelt, die für die Basisfunktion einer bestehenden Plattform mit hardwarenahen Bibliotheken genutzt werden. Die Anwendungsentwicklung auf OS Ebene wird erläutert mit den passenden Debugging und Analysemethoden. An Cloud Systemen und lokal eingesetzten Boards wird die Programmierung von Hardwarebeschleunigern gezeigt. Diese Hardwarebeschleuniger werden hinsichtlich vorteilhafter Implementierung und Optimierung besprochen.

Der Kurs vermittelt:

- Benutzung der Vitis IDE und Portierung aus Vivado SDK zur Vitis Plattform
- Entwicklung auf der Standalone Software Plattform
- Anpassung von Board Support Packages (BSPs) auf

der Xilinx Standalone library

- Debugging und Integration von eigenen Anwendungen
- Effektiver Einsatz von Gerätetreibern für die Xilinx Prozessoren
- Erstellen von Software Anwendungen mit dem OpenCL™ API für Hardware Kernels auf Alveo Accelerator Karten
- Ausführungssteuerung von OpenCL™ Anwendungen and the Xilinx runtime (XRT)
- Verständnis des Vitis Plattform Execution Model und XRT
- Erarbeitung der Kernel Entwicklung mit C/C++ und RTL
- Anwendung des Vitis Analyzer Tools auf Projektdaten
- Erklären Optimierungsstrategien für Accelerator Designs

Anwendbare Technologien

Architektur: Xilinx Alveo Accelerator Karten, SoCs und ACAPs

Voraussetzungen

Grundkenntnisse der Xilinx FPGA Architektur, Kenntnisse im Programmieren mit C/C++, Einsicht in den Software Entwicklungsprozess

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlicher Schulungsunterlagen sowie Pausengetränke und Mittagessen

Agenda

- | | |
|--|---|
| <ul style="list-style-type: none"> ■ Embedded Software Entwicklung mittels Embedded UltraFast Design Methodology ■ Zynq UltraScale+ MPSoC Architektur ■ Bedienung der Vitis Software Development Werkzeuge ■ System Debugger ■ Nutzen von Linker Scripts ■ Standalone Software Plattform Development und Interrupts ■ Operating Systeme: Einführung und Konzepte ■ Linux Software Anwendungsentwicklung und PetaLinux Tools ■ Erstellen einer Linux Anwendung mit der Vitis IDE ■ Überblick: Booting ■ Einführung in das Software Profiling | <ul style="list-style-type: none"> ■ Einblicke in Gerätetreiber ■ Basiswissen Hardware Beschleunigung ■ Alveo Data Center Accelerator Karten und Nimble Cloud ■ Vitis Execution Model und XRT ■ OpenCL Schnittstelle und Synchronization Techniques ■ Arbeiten mit NDRanges ■ Einführung in C/C++ basierte Kernel und RTL Kernel Wizard ■ Hardwarebeschleuniger ■ Optimierungsmethodik mit dem VITIS Analyzer Tool ■ Vitis Accelerated Libraries, BLAS, Fintech und OpenCV ■ Speicher Transfer Optimierung, XDMA und QDMA Datenfluss |
|--|---|



Workshop Versal ACAP System Architecture

Mit Versal stellt Xilinx die erste hochintegrierte Chip Serie aus der Technologiefamilie von ACAP - Adaptive Compute Acceleration Plattform bereit. Die Versal Chips weisen eine heterogene Architektur auf mit CPU Clustern, FPGA, High Speed Connectivity, Memory Management und AI Engines, die es ermöglichen, Software und DSP-Applikationen in Hardware zu parallelisieren und zu partitionieren, um die optimale Hardware-Architektur für spezifische Aufgabe zu nutzen. In diesem Workshop wird die Architektur vorgestellt, um die Bausteine optimal für anspruchsvolle Aufgaben nutzen zu können, ob Prozessor unterstützt, ob mittels AI-Engines, DSP-Engines oder programmierbare Logik (PL). Es werden die technischen Merkmale und Komponenten der Bausteine beschrieben, auf die mittels Vitis durch Software und Hardware-Kompilation spezifische Funktionen abgebildet werden können.

Viefältige Optimierungsmöglichkeiten bestehen, um Anforderungen von Performance, Reduktion von Verlustleistung, Verringerung von Latenzen bis hin zu Elementen funktionaler Sicherheit erfüllen zu können. Insbesondere dafür ist eine sinnvolle Partitionierung ein wichtiger Aspekt, der die Kenntnisse dieser Architektur voraussetzt. Entsprechend

werden die typischen ACAP Hard IPs wie Network-on-Chip (NoC) und die mehrfachen Memory Controller vorgestellt und im Kontext einer System Design Methodik vorgestellt. Diese stellt durchgängig den Plattform-Ansatz der Devices und der Toolchain in den Vordergrund und deckt die Aspekte der Plattform Erzeugung, System Integration und Simulation bzw. Validierung vor. Der Workshop eignet sich insbesondere für System Ingenieure und Entwickler in der Planungsphase von komplexen Aufgaben, wie z.B. in Cloud Anwendungen wie auch für Edge-Produktentwicklungen.

Anwendbare Technologien

Xilinx Versal ACAP Series

Voraussetzungen

Grundkenntnisse von Prozessor System Architekturen
Kenntnisse der C/C++ Programmierung
Grundkenntnisse der FPGA Architektur

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Einführung in Versal ACAP
- ACAP Architektur
- ACAP Adaptable Engines (PL)
- Processing System
- PMC, Boot and Konfiguration
- IO Konnektivität
- Clocking Architecture
- System Interrupts
- Timers, Counters, RTC
- DSP Engine und AI Engines
- Einführung in das Versal NoC
- Device Memory
- Serial Transceivers
- Einführung in PCIe, CPM and CCIX
- Security Merkmale
- Partitionierung von System-Architekturen

- Entwicklungsmethodik für Hardware, IP, und Plattform
- System Integration und Validation Methoden

Labs:

- Design Tool Flow
- Boot and Configuration
- High level introduction to IO and clocking resources
- Application development using the APU/RPU cores
- NoC Introduction
- System level simulation
- Versal ACAP Platform Creation



Workshop Compact Versal ACAP for HW Designer

Mit der Versal ACAP Familie (Adaptive Compute Acceleration Platform) erweitert Xilinx die Features und Flexibilität der bisherigen Bausteinfamilien. Hardware Entwicklern stehen zu den bekannten Vorteilen programmierbarer Logik leistungsfähige Prozessoren weitere Hard-IP zur Verfügung, um effektive em-bedded Systeme zu erstellen. Für Software Entwickler wird eine Vielzahl von Libraries und Methoden der Hardwarebeschleunigung für Prozessor-basierte Applikationen mit der Entwicklungsumgebung Vitis mitgeliefert.

Dieser Kurs vermittelt sowohl die Tool-, als auch die Architektur-spezifischen Aspekte, die für die Hardwareentwicklung auf Versal ACAPs relevant sind, was nicht nur die klassischen HDL Implementationen in PL betrifft. Es werden auch Einsichten in die Chip-Ressourcen geboten, wie Intelligent Engines und Peripherien, die mit dem Network on a Chip (NoC) mit hoher Bandbreite verbunden werden. Im Team basierten Arbeiten von Hardware- und Software Entwicklern wird im Kurs vermittelt, wie die Systemintegration mit den Tools von Vivado und Vitis erfolgt. Daher wird die Erarbeitung einer Hardware Plattform mit den IP Elementen der Versal ACAPs auf Basis der Vivado Design Tools vorgestellt und die Übergabe an den Vitis Software Development Flow gezeigt. Die Vitis Umgebung bietet die Tools für Debugging auf der Hardware und Einbindung der Treiber aller Hardware Feature, um damit ein effektives Einschalten der Hardware Designs zu ermöglichen. Ein weiterer Schwer-

punkt ist die Verknüpfung eigener Hardware Funktionen, die in der Regel in Händen des Hardwareentwicklers liegt. Hier gilt es, eine eigene Funktion mit AXI basierende Schnittstellen Technologie für optimalen Durchsatz und passende Latenz zu managen und diese auch für den Vitis Software Prozess zu verpacken. Um diese Embedded Designs in der Hardware verwenden zu können, schließt der Kurs mit der Darstellung der Generierung der Programmier-Files ab. Für tiefere Behandlung der Tools für Embedded Software Design auf der Versal ACAP Plattform bieten wir die Kurse „Compact Versal ACAP for SW Designers“ oder den übergreifenden „Professional Versal ACAP“ an.

Anwendbare Technologien

XILINX VERSAL ACAP Series

Voraussetzungen

Verständnis digitaler Systemarchitekturen

Grundkenntnisse des XILINX Vivado Flows

Grundkenntnisse der XILINX FPGA Architektur

Kenntnisse der Programmiersprache VHDL

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Overview of Embedded Hardware Development
- Versal ACAP Architecture Introduction
- Scalar Engines
- Adaptable and Intelligent Engines
- Driving the IP Integrator Tool
- NoC and Memory Introduction and Concepts
- Clocks, Resets Architecture
- IO Connectivity
- Overview of Embedded Software Development
- Driving the Vitis Software Development Tools

- Versal Interrupts: Hardware Aspects
- AXI Introduction and Variations
- AXI Transactions
- Connecting AXI IP
- Creating a new AXI IP
- Versal Programming Interfaces
- PMC, Boot and Configuration

Labs:

- Umfangreiche themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung



Workshop Compact Versal ACAP for SW Designer

Mit der Versal ACAP Architektur werden die Möglichkeiten des PS Systems aus der MPSoC Familie um leistungsfähige Komponenten erweitert und das Mapping von Applikationen auf den ARM Clustern der APUs (Application Processing Unit) und RPU (Realtime Processing Unit) und der Programmierbaren Lo-gik (PL) und weiterer IPs wie den AI Engines erleichtert. In diesem Kurs wird das Design einer Embedded Software vermittelt, das auf einer Plattform aus dem Vivado Design Flow aufsetzt.

Die XILINX Software Entwicklungsumgebung Vitis wird verwendet, um die Methoden der Softwareerstellung kennen zu lernen. Sowohl das Design von Software für einzelne Prozessoren, als auch die Verwendung von Operating Systems wie Linux wird gezeigt. Mit der Funktionsvielfalt der Versal Plattformen ist auf Systemebene die Partitionierung einzelner Funktionen auf die entsprechende Hardware wesentlich, um ein Design effizient umzusetzen. Daher ist auch die Analyse der Performance mit der Emulation oder im Target System Inhalt des Kurses. Um die vollen Möglichkeiten der mit Vitis bereitgestellten Lösungen zu zeigen, wird die Unterstützung verschiedener OS Ansätze dargestellt. Die Multicore-Hardware lässt den Parallelbetrieb mehrerer Betriebssysteme zu, wodurch Wissen um Synchronisierung, z.B. mittels OpenAMP Framework, wichtig für eine richtige Skalierung.

Es stehen dazu neben Linux auch Realtime Operating Systeme, FreeRTOS, als auch baremetal Implementationen zur Verfügung.

Die Versal Architektur bietet dynamische Power-Kontrolle, wozu mit dem PMC (Platform Management Controller) ein eigenes Target zur Verfügung steht. Daher wird im Kurs auf PMC Software Projekte eingegangen. Die spezifischen Toolflows für Accelerator-basiertes Design in Vitis, oder auch speziell auf den Versal AI Engine zielende Inhalte können unsere Kurse „Compact Vitis for Acceleration“ und die Versal AI Engine Kurs vermitteln.

Anwendbare Technologien

XILINX VERSAL ACAP Series

Voraussetzungen

Grundlagenkenntnisse von Prozessor Architekturen
Kenntnisse der C/C++ Programmierung
Grundlagenkenntnisse von FPGA Architektur

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Overview of Embedded Software Development
- Versal Architecture Overview
- Short Design Tool Flow Overview
- Versal ACAP Processing System
- Driving the Vitis Software Development Tool
- Debugging Methods
- Versal Application Partitioning
- Application Development and Debugging
- System Simulation
- FreeRTOS
- Symmetric MultiProcessing Linux
- Versal ACAP Software Build Flow: PetaLinux and Yocto
- Software Stack for Versal ACAP
- Deploying OpenAMP in a Heterogeneous System
- Versal Platform Controller (PMC)
- Power Management Controller
- Boot and Configuration
- Introduction to System Integration and Validation Methodology
- Outlook on Acceleration Toolflow

Labs:

- Umfangreiche themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung



Workshop Professional Versal ACAP

Dieser Power-Workshop kombiniert die Inhalte der PLC2 Workshops „Compact Versal ACAP for Hardware Designers“ sowie „Compact Versal ACAP for Software Designers“. Im Kurs wird das notwendige und tiefgreifende Wissen vermittelt die Architektur und den Workflow für die Versal ACAP Technologie zu verstehen und mit den Werkzeugen zu arbeiten. Für den Hardware Entwickler wird insbesondere der Vivado IP Integrator und der Vivado Workflow relevant, für den Software Entwickler die Vitis Toolsuite Umgebung, die ein große Bandbreite an Tools und Methoden liefert, um die heterogene Device Architektur möglichst effizient auszunutzen.

So werden umfassend die Tool-, als auch die Architekturspezifischen Aspekte für die Hardwareentwicklung als auch der Softwareentwicklung behandelt. Mit der Funktionsvielfalt der Versal Plattformen ist im System Design die Partitionierung einzelner Funktionen auf die entsprechende Hardware wesentlich, um ein Design effizient umzusetzen. Der Einsatz dazu notwendiger Tools zum Debugging und Deployment

auf der Hardware werden entlang dieser Diskussion auch betrachtet.

Siehe weitere Details auch in den Kursbeschreibungen von „Compact Versal ACAP for Hardware Designers“ und „Compact Versal ACAP for Software Designers“.

Anwendbare Technologien

XILINX VERSAL ACAP Series

Voraussetzungen

Verständnis digitaler Systemarchitekturen

Grundkenntnisse der Xilinx Vivado/SDK/Vitis Tools

Grundkenntnisse der Xilinx FPGA Architektur

Kenntnisse der Programmiersprachen C/C++

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Overview of Embedded Hardware Development
 - Versal ACAP Architecture Introduction
 - Scalar Engines
 - Adaptable and Intelligent Engines
 - Driving the IP Integrator Tool
 - NoC and Memory Introduction and Concepts
 - Clocks, Resets Architecture
 - IO Connectivity
 - Overview of Embedded Software Development
 - Driving the Vitis Software Development Tools
 - Versal Interrupts: Hardware Aspects
 - AXI Introduction and Variations
 - AXI Transactions
 - Connecting AXI IP
 - Creating a new AXI IP
 - Versal Programming Interfaces
 - Design Tool Flow Summary
 - Versal ACAP Processing System
 - Versal Application Partitioning
 - Debugging Methods
 - Application Development and Debugging
 - System Simulation
 - FreeRTOS
 - Symmetric MultiProcessing Linux
 - Versal ACAP Software Build Flow: PetaLinux and Yocto
 - Software Stack for Versal ACAP
 - Deploying OpenAMP in a Heterogeneous System
 - Versal Platform Controller (PMC)
 - Power Management Controller
 - Boot and Configuration
 - Introduction to System Integration and Validation Methodology
 - Outlook on Acceleration Toolflow
- Labs:**
- Umfangreiche themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung



Workshop Advanced Versal AI Engine

Innerhalb der Versal Adaptive Compute Acceleration Platform (ACAP) Familie führte Xilinx Versionen dieser Bausteine mit einer neuartigen IP ein, dem AI Engine. Diese AI Engines bieten hohe Rechenleistung und geringe Latenz für Datenverarbeitung mit hohen Anforderungen. Diese Schulung zeigt Programmieren, wie Funktions-Kernel für diese Versal AI Engines erstellt werden. Dazu werden zunächst auf die grundlegenden Elemente der Versal AI Engine vorgestellt, wie der VLIW Prozessor, die Schnittstellen und die Verbindungen zum Datenpfad und zur Speicherhierarchie im gleichförmigen Raster der AI Engines. Über die Vitis Toolchain werden auf den AI Engines Funktionen als Beschleuniger-Kernel abgebildet, die in C/C++ beschrieben werden. Hier wird gezeigt, wie solche AI Engine Kernels erstellt und verifiziert werden können. Dabei werden auch die Debugging Funktionen vorgestellt und die Vitis Analyzer-Tools angewendet, sowohl in Beispielen als auch in Labs. Für größere Signalverarbeitungsketten stellt Vitis ein API bereit, in dem mehrere Kernel mittels eines Dataflow Graph verknüpft werden, der dann komplett auf dem Versal AI Engine

Array abgebildet werden kann. Der Kurs zeigt, wie solche Graphen mit Vitis in das System Design eingebunden werden. Dabei wird auch auf die Partitionierung von Applikationen auf die verschiedenen heterogenen Kernel betrachtet. Um solche heterogenen Systeme zu erstellen, muss der Dataflow Graph mehrere Domänen, wie PL und die AI Engines verknüpfen. Die Methoden, diese Graph Knoten effizient zu verbinden, wie z.B. Streams werden erläutert und können anhand von Lab Sessions erarbeitet werden.

Anwendbare Technologien

XILINX VERSAL ACAP AI Series

Voraussetzungen

Grundkenntnisse von Embedded Controller

Einblicke in die Vitis Toolchain

Kenntnisse in der Programmiersprache C/C++

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Überblick über die Versal ACAP Architecture
- Einführung in die Architektur des Versal AI Engine
- Grundlagen des Versal AI Engine Memory und Datentransports
- Versal AI Engine Tool Flow mit Vitis
- Application Partitioning auf Versal ACAPs
- Datentypen und intrinsische Funktionen
- Window und Streaming Data APIs
- Das Programming Model: Single Kernel
- Adaptive Data Flow Graphen
- Das Programming Model: Multiple Kernels
- System Design Flow mit Vitis
- Einführung in das Debugging mit AI Engine Kernels
- Vorstellung Versal AI Engine DSP Library

Umfangreiche themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung



Workshop Expert Versal AI Engine

Innerhalb der Versal Adaptive Compute Acceleration Plattform (ACAP) Familie führte Xilinx Versionen dieser Bausteine mit einer neuartigen IP ein, dem AI Engine. Diese AI Engines bieten hohe Rechenleistung und geringe Latenz für Datenverarbeitung mit hohen Anforderungen.

Dieser Kurs vermittelt die Beschleunigung von Applikationen mit C/C++ Kernel auf den Versal AI Engines. Dazu werden zunächst auf die grundlegenden Elemente der Versal AI Engine vorgestellt, wie der VLIW Prozessor, die Schnittstellen und die Verbindungen zum Datenpfad und zur Speicherhierarchie im gleichförmigen Raster der AI Engines. Die Kernel für die AI Engines werden mit der Vitis Toolchain in C/C++ programmiert. Der Kurs zeigt, wie diskrete AI Engine Kernel mit den Intrinsic Functions des API in der Vitis Toolchain erstellt werden können.

Mit diesem Verständnis der Kernel Programmierung zeigt der Kurs, wie diese Kernel in einem Dataflow Graph eingebunden werden. Anhand der Dataflow Graphen wird auch das Data Movement zwischen den verschiedenen Compute Domains, wie den AI Engine und dem NoC und der programmierbaren Logic (PL) erläutert und die Interfaces mit

den jeweiligen Eigenschaften erklärt. Der Kurs vermittelt, wie solche Designs mit den Tools der Vitis Plattform analysiert und optimiert werden können.

Um solche heterogenen Systeme zu erstellen, muss der Dataflow Graph mehrere Domänen, wie PL und die AI Engines verknüpfen. Die Methoden, diese Graph Knoten effizient zu verbinden, wie Streams, Cascade Streams, Buffer Location Constraints, Run-Time Parametrierung und die passenden APIs werden erläutert und anhand von Lab Sessions erarbeitet. Die so optimierten Signalverarbeitungsketten werden in ein Versal System Design eingebettet.

Anwendbare Technologien

XILINX VERSAL ACAP AI Series

Voraussetzungen

Grundkenntnisse von Embedded Controller

Einblicke in die Vitis Toolchain

Kenntnisse in der Programmiersprache C/C++

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Überblick über die Versal ACAP Architecture
- Einführung in die Architektur des Versal AI Engine
- Versal AI Engine Memory und Datentransport
- Versal AI Engine Tool Flow mit Vitis
- Application Partitioning auf Versal ACAPs
- Datentypen: Skalare and Vektordatentypen
- Intrinsic Funktionen
- Window and Streaming Data APIs
- Das Programming Model: Single Kernel
- Adaptive Data Flow Graphen
- Advanced Graph Input Specifications
- Das Programming Model: Multiple Kernels
- Application Partitioning Walkthrough
- System Design Flow mit Vitis
- Einführung in das Debugging mit AI Engine Kernels
- AI Engine Application Debug und Trace.
- Einblicke in die AI Engine Kernel Optimierung
- Fortgeschrittene intrinsic Funktionen
- Vorstellung der Versal AI Engine DSP Library

Workshop AXI Interface Technology

Der Interface Standard ARM-AXI hat sich im ASIC, FPGA, SoC, MPSoC Systemen weitgehend durchgesetzt, so dass für nahezu alle Cores diese Interface Technologie zur Verfügung steht, wie auch für User-Peripherals.

Aber nicht nur Busstandards wie im Kontext der Prozessoren wie MicroBlaze oder ZYNQ PS, auch DSP Cores oder rudimentäre Hardware wie FIFOs können als AXI konforme Schnittstelle eingerichtet werden.

Dieser Workshop vermittelt fundamentales Wissen um den Standard und Methoden, eigene Hardware Komponenten AXI-konform entwickeln und testen zu können. Auch erweitert sich die AXI IP-Palette der Drittanbieter, da viele Technologieanbieter bei der Standardisierung mitwirkten.

Die AMBA Protokolle werden aufgezeigt und inwieweit AXI für FPGA Implementierungen optimale Konfigurationen der Signalkommunikation erlaubt. Die Architektur der AXI Komponenten performant oder

ressourcenschonend zu gestalten, erfordert Fachwissen, da viele Faktoren zu berücksichtigen sind.

Mit praxisnahen Übungen begleitet vermittelt der Workshop Protokollverständnis und Expertenwissen der Anwendung.

Anwendbare Technologien

XILINX FGAs, SoCs, MPSoCs, RFSoc & ACAPs

Voraussetzungen

Kenntnis der FPGA Technologie u. XILINX Design Tool Flow
Grundkenntnisse VHDL

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

AXI Introduction

- ARM AMBA
- AXI3 and AXI4 Standards
- AXI4 System Architecture

AXI Variations

- AXI Memory Mapped
- AXI Streaming
- AXI Coherency Extension (ACE)
- AXI Coherent Interfaces (Zynq US+ MPSoC)

AXI Transactions

- AXI Handshaking
- AXI Read and Write Timing
- AXI Master and Slave Transaction

AXI IP Management

- AXI Connecting AXI IP
- AXI Related Cores
- AXI Clock Domain Crossing
- AXI Protocol Checkers

AXI Streaming

- AXI Streaming Introduction
- AXI Streaming FIFO
- AXI Streaming DMA

AXI User IPs

- Creating a New AXI IP with the Wizard
- Packaging AXI IP
- AXI Support with IPI Block Automation
- AXI Support with Xilinx System Generator
- AXI Support with Vivado HLS

AXI IP Verification

- AXI BFM Simulation Using Verification
- JTAG to AXI Master Core
- Debug Flow in an IP Integrator Block Design
- AXI Monitor und ChipScope ILA

AXI Performance Analyzing

- AXI DMA
- AXI Traffic Generator (ATG)
- AXI Performance Monitor (APM)
- System Performance Monitoring (SPM)

Labs

- Exploring AXI Transactions Using the ATG Core
- Building Custom AXI IP
- Introduction to Verification IP Simulation
- JTAG to AXI Master Core Demo Script
- Exploring AXI Performance
- Configuration of NoC in Versal

Workshop High-Speed Ethernet – Hands-On System Development

Practical Design Experience for You!

Ohne Ethernet ist moderne Kommunikationstechnik nicht denkbar. Ethernet ist ein wichtiger Bestandteil in vielen Anwendungsbereichen und ist von der IEEE für die Informationsübertragung über diverse physikalische Medien als auch für unterschiedliche Übertragungsgeschwindigkeiten standardisiert. Ethernet wird gleichermaßen in LANs als auch WANs verwendet und ist durch seine Verfügbarkeit und vergleichsweise hohe Übertragungseffizienz ideal für spezifische Lösungen in vielen Bereichen.

Ziel des 3-tägigen PLC2 Workshops ist die Befähigung zur eigenständigen Entwicklung von Ethernet-basierten Kommunikationsprotokollen. Das notwendige Ethernet Basiswissen und darauf aufbauende wichtige Netzwerkprotokolle, wie z.B. die Internet Protokoll Familie wird anhand des OSI Referenzmodells vermittelt und vertieft.

XILINX spezifische Ethernet Lösungen und deren Partitionierung in PS/PL Komponenten werden diskutiert und im Zusammenhang mit verfügbaren XILINX DMA Lösungen bewertet. Dabei wird auch auf das Management externer Ethernet PHYs eingegangen.

Ein Ethernet basiertes Kontrollprotokoll für einen Daten-

generator wird während des Workshops spezifiziert, entwickelt und auf bereitgestellten Evaluationsboards getestet. Die Leistungsfähigkeit möglicher Systemdesigns werden untersucht und gegenübergestellt. Dabei werden auch Debugging Möglichkeiten bei auftretenden Herausforderungen vorgestellt. Das in dem Workshop entwickelte Kommunikationsprotokoll kann als Basis für weiterführende remotefähige Anwendungen dienen.

Anwendbare Technologien

XILINX FPGA und Derivate

Voraussetzungen

Gute Kenntnisse der XILINX FPGA Architekturen

Grundlegende VHDL und C Kenntnisse

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- OSI Model
- Ethernet Protocol Basics
 - Ethernet/MAC Packets
 - Address Resolution Protocol (ARP)
- IP Basics
 - Internet Protocol (IP)
 - Internet Control Message Protocol (ICMP)
 - User Datagram Protocol (UDP)
 - Transmission Control Protocol (TCP)
- Ethernet Physical Interfaces
 - Media Independent Interfaces (MIIs)
 - Media Dependent Interfaces (MDIs)
 - Line Coding (8B/10B, 64B/66B)
 - Eye Diagram
- Ethernet PHY control
- Types of XILINX Ethernet Cores and Their Usage
- XILINX DMA Solution
 - (Dis-)Advantages
 - Configuration
- XILINX Ethernet Driver
- PS/PL Partitioning
- Client Server App
- Custom Protocol
 - Definition
 - Implementation
 - Verification
- Performance Comparison
- Ethernet Communication Debugging
 - Wireshark Debugging
 - Loopback Modes
 - BIST
- **Lab:** Ethernet MAC Setup (PS, PL)
- **Lab:** Using XILINX Ethernet Driver in SDK
- **Lab:** External PHY Configuration and Control
- **Lab:** Loopback Modes
- **Lab:** Debugging
- **Lab:** Server-side Data Generator Integration
- **Lab:** Custom Protocol Implementation in SW
- **Lab:** Custom Protocol Implementation in HW with SW Management

Workshop

PCI Express Hands-on System Development

Seit seiner Einführung im Jahre 2002 durchdringt PCI Express als Hochgeschwindigkeitsschnittstelle immer mehr die DSP und Mikrocontrollerwelt und hat sich zu einem de-facto Standard entwickelt. Viele Anwendungen nutzen hierbei die Direct Memory Access (DMA) Eigenschaften von PCI Express, um Daten schnell zum oder vom CPU Speicher zu streamen.

Dieser Kurs vermittelt die Grundlagen und wichtiges Praxis Know-How für den Aufbau und die Inbetriebnahme von hochperformanten DMA PCI Express Streaming Endpoints mit XILINX FPGAs.

Der Teilnehmer lernt grundlegende PCI Express Protokolleigenschaften kennen, die den Datendurchsatz bei Streaming Applikationen negativ beeinflussen und bei der Konzeption frühzeitig berücksichtigt werden müssen. Es wird auf verschiedene DMA Architekturen eingegangen und diese mit ihren Vor- und Nachteilen gegenübergestellt. Anhand

einer Praxiseinheit werden am Beispiel eines PCIe Traffic-Generators praktische Durchsatzvermessungen durchgeführt und systemrelevante Kenngrößen (CRC Fehler, Stall Time) vermessen.

In weiteren Einheiten wird auf Simulationsaspekte, Leiterplattenthemen und Debugging Möglichkeiten eingegangen. Der Workshop ist praxisnah und enthält Übungen, die helfen, das theoretische Wissen direkt anwenden zu können..

Anwendbare Technologien

XILINX 7 Series, alle Ultrascale Architekturen

Voraussetzungen

Grundkenntnisse VHDL, Grundkenntnisse im XILINX Design Tool Flow, Grundkenntnisse FPGA-Technologie

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction to PCI Express

- PCIe Systemarchitecture
- PCIe specific terms
- Negotiation

Overview Protocol Specification

- TLPs (Requests & Completions)
- BARs
- Configuration Space

Performance limiting influences

- Understanding Flow Control
- PowerManagement
- Measuring Performance
- Detecting CRC Errors
- **Lab:** Traffic Generator

Booting and Configuration

- Reaching the 100 ms boottime requirement
- TandemPROM / TandemPCIe
- Field Updates with 7 Series FPGAs

PCB Design Considerations

- Highspeed Lanes
- Clocking
- Layout aspects

Simulation

- Designing PCIe Testbenches
- How to accelerate Simulation
- **Lab:** Simulating and optimizing simulation time

Debugging a PCI Express System

- IBERT Eyediagrams
- Chipscope
- Completion Timeouts
- **Lab:** Exploring a PCI Express Link with IBERT and Chipscope

DMA architectures

- Contiguous DMA Memory
- Scatter/Gather DMA Memory
- Building Descriptor Lists



Workshop Designing with RF Data Converters

Die neue XILINX Zynq UltraScale+ RFSoc Familie erlaubt schnelle Interfaces mit integrierten Analog/Digital und Digital/Analog Wandlern.

Dieser 3-tägige Workshop beginnt mit einer allgemeinen Beschreibung der RFSoc Familie. Es werden die Hauptkomponenten erläutert und einige interessante Applikationsbeispiele gezeigt.

Die RF-ADC- und RF-DAC Wandler werden ausführlich diskutiert, einschließlich deren Architektur und Funktionalität. Der Teilnehmer erwirbt das notwendige Wissen um die Wandler zu konfigurieren, zu simulieren und zu implementieren.

Besondere Herausforderungen bestehen im PCB Design. Dazu werden die Anforderungen seitens der RFSoc Bausteine vorgestellt und Realisierungsoptionen ausführlich diskutiert.

Zusätzlich geht dieser Kurs auf den SD-FEC Hard-IP ein.

Auch hier werden die notwendigen Schritte zur Konfiguration und Simulation behandelt.

Anwendbare Technologien

Zynq UltraScale+ RFSoc

Voraussetzungen

Grundlegende Kenntnisse über Zynq UltraScale+ MPSoC Hardware- und Software-Design

Grundlegende Kenntnisse über D/A- bzw. A/D Wandler (Begriffe und Prinzipien)

Grundlegende Kenntnisse über Forward Error Correction (Begriffe und Prinzipien)

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inclusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Zynq UltraScale+ RFSoc Overview

- RF backgrounder
- Describing RFSoc architecture overview
- Data converter solutions overview
- SD-FEC solutions overview
- Tool support

RF-ADC

- Recover RF-ADC basic vocabulary
- Describing the RF-ADC architecture, functionality, and interfaces
- Configure RF-ADCs
- Utilize RF-ADCs in IP Integrator
- Software driver overview

RF-DAC

- Recover RF-DAC basic vocabulary
- Describing the RF-DAC architecture, functionality, and interfaces
- Configure RF-DACs
- Utilize RF-DACs in IP Integrator
- Software driver overview

RF Data Converter Design

- Describe common features
- Applying the design flow
- Utilize the example design by simulation and implementation

PCB Design Topics for RFSoc Devices

- Describe the power requirements
- Performing power estimation
- Utilize the power design
- Describe the analog signal requirements
- Understanding PCB material and layer stackup options
- Utilize the analog trace design

RFSoc Soft-Decision FEC

- Recover error correction basic vocabulary
- Understanding the SD-FEC architecture, functionality, and interfaces
- Configure SD-FEC IPs
- Utilize SD-FEC by simulation
- Software driver overview

Labs

- RF-ADC IP Generation
- RF-DAC IP Generation
- RF-ADC Simulation
- RF-DAC Simulation
- RF Data Converter Implementation
- RF Data Converter Power Estimation
- SD-FEC IP Generation
- SD-FEC IP Simulation
- Evaluation Tool, RF-Analyzer, PYNQ



PowerWorkshop Professional RFSoc

Die neue XILINX Zynq UltraScale+ RFSoc Familie erlaubt schnelle Interfaces mit integrierten Analog/Digital und Digital/Analog Wandlern.

Dieser 5-tägige Workshop kombiniert eine umfängliche Diskussion der Hard- und Softwareaspekte zum Design mit der RFSoc Familie.

Die RF-ADC und RF-DAC Wandler werden ausführlich erläutert, einschließlich deren Architektur und Funktionalität. Der Teilnehmer erwirbt das notwendige Wissen um die Wandler zu konfigurieren, zu simulieren und zu implementieren.

Besondere Herausforderungen bestehen im PCB Design. Dazu werden die Anforderungen seitens der RFSoc Bausteine vorgestellt und Realisierungsoptionen ausführlich diskutiert.

Seitens des Processing Systems wird anfangs ein Gesamtüberblick vermittelt, um dann in folgenden Abschnitten die ARM Architekturen mit APU und RPU sowie XILINX Erweiterungen, wie die Platform Management Unit (PMU) zu besprechen. Begleitende Übungen ermöglichen den Teilnehmern mit Anwendungsbeispielen die Prozessor-Systeme

zu konfigurieren und Software Applikationen zu simulieren und in der Hardware zu starten. Es werden systemspezifische Merkmale, wie System Protection sowie Takt- und Resetstrukturen besprochen. Ein wichtiger Abschnitt dient auch der DRAM Speicherkonfiguration mit Multi-Port Management, Quality-of-Service und den erforderlichen AXI Schnittstellen, um die hohen Bandbreiten der ADC/DAC Interfaces auf DRAM Speicher zu verwalten.

Anwendbare Technologien

Zynq UltraScale+ RFSoc

Voraussetzungen

Grundlegende Kenntnisse über Zynq UltraScale+ MPSoC

Hardware- und Software-Design

Grundlegende Kenntnisse über D/A- bzw. A/D Wandler (Begriffe und Prinzipien)

Grundlegende Kenntnisse über Forward Error Correction (Begriffe und Prinzipien)

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Processing System Hard- and Software

- PS Overview
- APU and RTU
- Software Environments
- HWSW Virtualization
- Power Management
- System Coherency and Protection
- Clocks and Resets
- DDR and QoS
- Boot and Configuration
- Drivers
- Software Developmen
- Ecosystem Support

Zynq UltraScale+ RFSoc Overview

- RF backgrounder
- Describing RFSoc architecture overview
- RF data converter solutions overview
- SD-FEC solutions overview

RF-ADC/DAC

- Recover ADC/DAC basic vocabulary
- Describing the RF-ADC/DAC architecture, functionality, and interfaces
- Configure RF-ADCs/DACs
- Utilize RF-ADCs/ DACs in IP Integrator

RF Data Converter Design

- Describe common features
- Applying the design flow
- Utilize the example design by simulation and implementation

PCB Design Topics for RFSoc Devices

- Describe the power requirements
- Performing power estimation and design
- Describe the analog signal requirements
- Understanding PCB material and layer stack-up options
- Utilize the analog trace design

RFSoc Soft-Decision FEC

- Recover error correction basic vocabulary
- Understanding the SD-FEC architecture, functionality, and interfaces
- Utilize SD-FEC design

Labs

- Various labs deepen the knowledge and allowing practice oriented training
- Using ZCU111 hardware
- Practicing RF data converter evaluation and debugging

PowerWorkshop Professional MicroBlaze System Design

Der 5-tägige PLC2-PowerWorkshop „Professional MicroBlaze System Design“ wendet sich an embedded Hardware- und Softwareentwickler,

die mit der Prozessorarchitektur XILINX MicroBlaze arbeiten möchten. Der MicroBlaze Prozessor-Core kann in der Hardware sehr flexibel konfiguriert werden, um diesen je nach Anforderung, wie leistungsoptimiert, frequenzoptimiert, Ressourcen-optimiert und auch mit individuellen Anforderungen im FPGA multifunktionell einzusetzen.

Im IP-Integrator der XILINX Vivado Tools Suite wird die Hardware Plattform im FPGA individuell konfiguriert: Prozessor, Speicher intern, Speicher extern, Peripherien, Debug-Unterstützung mit vielen Konfigurationen wie Taktung, Wortbreite, Interrupt/Polling, DMA, und Feature der IP-Komponenten.

Auch Multi-Prozessor Lösungen sind möglich.

Der erste Teil des Workshops widmet sich der Erstellung der Embedded Plattform unter Vivado und der zweite Teil der Software Entwicklung mit XILINX Vitis.

Zahlreiche praktische Übungen bieten einen optimalen Einstieg für Ihr MicroBlaze Projekt.

Anwendbare Technologien

Xilinx FGAs, SoCs, MPSoCs, RFSocCs & ACAPs

Voraussetzungen

VHDL und FPGA Architekturkenntnisse
Grundlagen der Programmiersprache C

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Embedded Processor Design

- Vivado Design Environment
- IP Integrator
- Downloading Methods

Hardware Bus Architectures

- Arbiter/Master/Slaves
- AMBA/AXI Interfaces

Architecture and Configurations

- MicroBlaze Architecture
- MicroBlaze Processor Configurations
- Arm Cortex-M1/M3 versa MicroBlaze

Memories

- BlockRam / DDRAM / Flash

Designing Custom Peripherals

- IP Integrator and IP Packager
- Bus Functional Model (BFM)
- IP Integration

Software Development Using Vitis

- Board Support Packages (BSP)

Address Management

- Software Interrupts
- Interrupt and Exception Handler

Software Platform Download and Boot

- Application Debugging
- Application Profiling
- Bootloader

Writing a Custom Device Driver

Advanced Services and Operating Systems

- XILINX Libraries
- Ethernet - LwIP Stack

Embedded Projekt Management

- Project Management in Vivado

Übungen

- MicroBlaze Hardware Design
- Driving the Vitis Tool Environment
- Adding IPs in the Vivado Project
- Building Custom AXI IP
- BFM Simulation for AXI Peripherals
- Integrating a Custom Peripheral
- Linker Script
- Application Development
- Software Interrupts
- Debugging
- Software Profiling Overview
- Writing a Device Driver
- Designing with ARM Cortex-M1

Workshop Developing Multimedia Solutions with the VCU and GStreamer

Die Zynq UltraScale+ MPSoC EV Derivate mit integriertem Video Codec Unit (VCU) bieten für komplexe Multimedia Verarbeitungen eine Unterstützung des GStreamer Frameworks.

Dieser Kurs zeigt, wie die Verwendung der Video-Codec Einheit VCU dazu beiträgt, eine optimierte Performance zu erzielen, indem zeitkritische Aufgaben auf das dedizierte Modul ausgelagert werden. So wird auch eine Übersicht von Xilinx Multimedia Lösungen vermittelt, um insbesondere Video-/Audio Applikationsentwicklungen mit der VCU zu ermöglichen.

So wird auch die Architektur des VCU Encoders/Dekoders dargestellt und wie diese über das APU System DMA unterstützt zu steuern ist. Für diese Technologie bietet Xilinx einen Software Stack für Multimedia Applikationsentwicklungen, der ebenfalls erläutert wird.

Insbesondere die Verwendung des GStreamer-Frameworks

aus dem Software-Stack zum Erstellen verschiedener Multimedia-Pipelines ermöglicht eine höher effizientere Echtzeit Multimedia Entwicklung.

Anwendbare Technologien

Xilinx Zynq UltraScale+ MPSoC (EV)

Voraussetzungen

Grundkenntnisse der Video Technologie

Grundkenntnisse der Video Codec Unit (VCU)

Grundkenntnisse von Xilinx Zynq MPSoC

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Multimedia Overview
- Xilinx Multimedia Solutions
- Zynq UltraScale+ MPSoC: Multimedia Blocks
- Introduction to the Video Codec Unit (VCU)
- Zynq UltraScale+ MPSoC: VCU Architecture
- VCU-supported Standards and VCU Latency and Performance
- VCU Software Stack
- Introduction to the GStreamer Framework
- Video-supported Frameworks in Linux: V4L2, DRM, KMS
- Multimedia Connectivity and Processing IPs
- Streaming Pipeline Using GStreamer
- Xilinx Audio and Graphics Solutions
- Xilinx Targeted Reference Designs Overview

Übungen

- Zynq UltraScale+ MPSoC: VCU Architecture
- Introduction to the GStreamer Framework
- Video-supported Frameworks in Linux: V4L2, DRM, KMS
- Streaming Pipeline Using GStreamer

Workshop

Compact Embedded Linux

Beim Einsatz von FPGA-Designs im Zusammenspiel mit einem prozessorbasierten System bietet sich Linux als offenes Betriebssystem an. Doch wie erstelle ich ein Embedded-Linux System für meine Hardware-Plattform wie z.B. für den XILINX ZYNQ SoC? Von der Entwicklungsumgebung über den Boot-loader und dem Linux Kernel mit Root-Filesystem werden alle Komponenten eines Embedded-Linux-Systems diskutiert, angepasst und verwendet. Dabei kommen nur freie Werkzeuge wie GNU Compiler, u-boot oder BusyBox zum Einsatz. Hierdurch erlernt der Teilnehmer nicht nur wie ein Embedded-Linux-System erstellt wird, sondern er ist auch in der Lage das Linux System für sein eigenes Projekt anzupassen.

Dieser Kurs ist praktisch orientiert, so dass Abschnitte durch Übungen gefestigt werden. Die verwendete Zielplattform ist der XILINX ZYNQ, das erlernte Wissen lässt sich jedoch auch auf andere Technologien wie z.B. Prozessorsystemen anwenden.

Anwendbare Technologien

Prozessor basierende Embedded-Linux Systeme wie XILINX ZYNQ und andere.

Voraussetzungen

Erfahrungen mit dem Umgang mit Linux als Betriebssystem sind wünschenswert, jedoch nicht notwendig. Es werden jedoch in diesem Training die wesentlichen Linux Grundlagen für Hardware Entwickler diskutiert. Des Weiteren sollten die Teilnehmer grundlegendes Wissen über Makefiles und C-Programmierung mitbringen.

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Linux Basics for Hardware developers

- Linux Root filesystem concept
- Using the Console
- Scheduler, Processes and tasks
- Cross compiler and libraries
- Patching

Boot loader

- General boot process
- U-boot configuration and compilation

- Hardware debugging with u-boot
- Network boot for practical development

Linux kernel and root file system

- Linux kernel architectural basics
- Boot process
- Flatten Device Tree support
- Linux folder hierarchy
- File systems and wear leveling
- C library glibc, eglibc, uclibc

Workshop Embedded Linux Development with Yocto Project

In immer mehr FPGA/SoC Systemen wird Linux als Betriebssystem eingesetzt. Hardware Entwickler müssen deshalb Linux Systeme erstellen oder für die eigenen Bedürfnisse anpassen. Dies kann jedoch sehr viel Zeit in Anspruch nehmen, wenn das Linux System von Grund auf erstellt werden muss. In der Vergangenheit wurde hier oft ein bestehendes Linux System angepasst in dem nicht benötigte Elemente entfernt, und zusätzliche Software installiert wurde. Dieser Ansatz hat jedoch den Nachteil, dass Entwickler nie 100% wissen was eigentlich in dem Linux System an Komponenten vorhanden ist, was zu lizenzrechtlichen Problemen führen kann. Mit dem Yocto Project Build System sieht dies ganz anders aus. Mit diesem Tool sind Sie in der Lage eigene Linux System von Grund auf zu erstellen. Sie haben somit immer den genauen Überblick der verwendeten Komponenten. Des Weiteren erlaubt das Yocto Project Schichtenmodell dasselbe Rootfilesystem für verschiedenste Hardware Plattformen zu kompilieren, und macht es sehr leicht weitere Systemkomponenten von Drittanbietern in das Linux Image zu integrieren. In diesem Training lernen die Teilnehmer wie

sie ein eigenes Linux System mit dem Yocto Project Build System aufbauen können. Als Hardware Plattform wird ein XILINX ZYNQ basierendes Entwicklungsboard verwendet. Es wird gezeigt wie Board Support Packages und Meta-Layer aufgebaut sind und in das Buildsystem integriert werden. Am Ende des Trainings sind die Teilnehmer in der Lage die Yocto Project Tools anzuwenden, um ein eigenes Linux System zu generieren.

Anwendbare Technologien

Prozessor basierende Embedded-Linux Systeme wie XILINX ZYNQ und andere.

Voraussetzungen

Teilnehmer kennen den Aufbau und die Komponenten eines Embedded Linux Systems. Wissen, wie es in dem Training Compact Embedded Linux vermittelt wird ist als Grundlage notwendig.

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Installation and Configuration

BitBake Tool Basics

- Image generation
- Recipes and Append files
- Classes
- Configuration files and Patches

Recipe Design

- Creating custom recipes
- Adapting existing recipes
- Building root filesystem recipes

Board Support Packages

- Configuration
- Bootloader, Kernel and device tree
- Graphical Interfaces
- Hob & Toaster

Generating SDK

- Cross compiler
- Eclipse Plugin

Yocto Project Licensing features

Workshop

Embedded Linux Treiberentwicklung

Embedded Linux ist heute ein wichtiger Bestandteil von FPGA Systemen. Moderne System-On-Chip Lösungen wie der ARM basierende XILINX ZYNQ SoC, erlauben es schnelle FPGA Hardware mit der Flexibilität eines Linux Betriebssystems zu kombinieren. Damit Linux Applikationen auf die implementierten FPGA IP Cores zugreifen können, müssen entsprechende Linux Treiber erstellt werden. In diesem Training werden zuerst alle benötigten Linux Kernel Grundlagen erarbeitet, die dann im Weiteren für die Implementierung eines eigenen Treibers verwendet werden. Nach diesem Seminar kennen die Teilnehmer wichtige Aspekte des Linux Kernels und sind in der Lage eigene Treiber zu entwickeln, oder vorgegeben Treiber anzupassen. In diesem Kurs wird ein Embedded-Linux System mit einem FPGA IP Core für einen XILINX ZYNQ vorgegeben. Für dieses System wird in den Übungen ein eigener Treiber entwickelt. Es werden alle Themen von der Hardware-Definition, über Schutzmechanismen und Timing bis zum Debugging behandelt. Dieser Kurs ist praktisch orientiert, so dass je-

der Abschnitt durch Übungen abgeschlossen wird. Das Hauptaugenmerk bei dieser Schulung liegt auf Charakter- Treibern, Plattform Treibern und dem Sysfs interface. Weiterhin werden die grundlegenden Elemente der Linux DMA API besprochen.

Anwendbare Technologien

Prozessor basierende Embedded-Linux Systeme wie XILINX ZYNQ und andere.

Voraussetzungen

Kenntnisse wie sie im Kurs „Compact Embedded Linux“ vermittelt werden sind vorausgesetzt. Teilnehmer können sich auf der Linux-Shell bewegen, Makefiles verwenden, C-Programme erstellen und ändern. Grundlegendes Wissen über Prozessor- und FPGA-Hardware sind empfehlenswert.

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Linux Driver Basics

- Virtual, logical and physical addressing
- Process contexts
- Makefiles for driver module development
- Character-, Block, and Network-Devices

Hardware Access

- Device Tree basics
- Registers access and interrupts
- Clock management

Protection Mechanisms

- Race conditions
- Mutex and Semaphores
- Spinlocks

Timing and Working

- Kernel Threads, Tasklets and work queues
- Jiffies, high resolution timers and wait queues

Driver Debugging

- Printk, oops message analysis with gdb
- procfs and sysfs virtual file systems for driver debugging
- Kernel Debugger

Workshop Embedded Design with PetaLinux Tools

XILINX bietet mit PetaLinux eine Skript-basierte Open-Source Linux Build Lösung sowohl für ZYNQ Derivate wie auch FPGAs mit einem MicroBlaze Core, ergänzt mit kommerziellem Support, Dokumentation, Wartung und Pflege, wo im Open-Source Kontext große Lücken klaffen.

Dieser Workshop vermittelt anfangs Grundlagen und Anwendung eines Embedded Linux Betriebssystems.

PetaLinux bietet eine einfachere Form der Kernel Konfiguration basierend auf Shell Skripte, einfachere Methoden, QEMU Simulation, Firmware Packing, Templates sowie die Dokumentation. Die Pflege von Open-Source basierenden Linuxsystemen in der XILINX Toolchain ist mit PetaLinux von großem Vorteil.

Auch wird es Linux Einsteigern leicht gemacht, den Kernel zu konfigurieren, die Hardwareperipherie zu unterstützen und das Embedded System schnell und effizient mit User-Applikationen zu entwickeln.

Auch das Einbinden von Hardware Peripherien mit notwendiger Treiberunterstützung wird in Grundlagen behandelt.

Mit der Teilnahme an diesem Workshop sind Sie in der Lage, mit einem Embedded Linux System zu arbeiten.

Anwendbare Technologien

XILINX FPGA, SoCs, MPSoCs, RFSocS and ACAPs

Voraussetzungen

Grundkenntnisse Embedded Systems Software Design

Grundkenntnisse der Programmiersprache C

Grundkenntnisse von Linux sind von Vorteil

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Embedded Linux Overview

Embedded Linux Components

Introduction to the PetaLinux Tools

Application Development and Debugging

Networking and TCP/IP

Device Drivers, User Space I/O, and Loadable Modules

Custom Hardware Development and Interfacing

Board Bring Up with the Vivado Design Suite and PetaLinux Tools

Device Drivers for the Hardware

Übungen

- **Lab:** A First Look
- **Lab:** Build and Boot an Image
- **Lab:** Application Development and Debug
- **Lab:** Networking and TCP/IP
- **Lab:** Accessing Hardware Devices from User Space
- **Lab:** Basic Hardware Design with the Vivado and PetaLinux Tools
- **Lab:** Custom Hardware Development
- **Lab:** Custom Driver Development



Workshop Compact Python for Embedded

Im Bereich Embedded gilt seit langem die Programmiersprache C als der Standard. Komplexere Anwendungen und schnellere Time-to-Market-Anforderungen verlangen jedoch nach Alternativen. Python wurde traditionell zwar eher für Web und Desktop Applikationen verwendet, bietet aber dank seiner Unterstützung für C/C++ Bibliotheken und der Bereitstellung von Frameworks zur Verarbeitung von komplexen Algorithmen eine solche Alternative. XILINX bietet ein Open-Source Framework (PYNQ), welches Python erlaubt mit dem Zynq-Portfolio zu interagieren. Dadurch können komplexere Anwendungen aus den Bereichen Analyse großer Datenmengen, maschinellem Lernen (ML) oder künstlicher Intelligenz (KI) die Vorteile der schnelleren Verarbeitung durch die programmierbare Logik ziehen. Realisiert wird dies durch hybride Bibliotheken, welche eine neue Form von Bibliotheken darstellt. Eine solche hybride Bibliothek beinhaltet sowohl einen Bitstream

und den damit verbundenen hardwarenahen C-Code, aber auch eine Programmier-Schnittstelle (API) für Python. In diesem Kurs lernen Sie die Grundlagen in der Programmierung mit Python, welche Sie in einem Workshop anwenden. Sie sind anschließend in der Lage, Prototypen auf Basis von Python auf embedded Systemen in kurzer Zeit zu entwickeln.

Anwendbare Technologien

Python, PYNQ

Voraussetzungen

Grundlegende Kenntnisse über objektorientierte Programmierung

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Einführung

Erste Schritte

- Environment Setup
- Gründe für Python
- Einsatz von Jupyter

Programmierung mit Python

Grundlagen

- Variablen
- Typen, I/O und Import
- Operatoren & Namespace
- Datentypen

Abläufe

- Statements
- Loops

Funktionen

- Function & Argument
- Rekursion
- Modul & Package

Objekte und Klassen

- OOP
- Vererbung

Embedded Hardware

Einführung

- PYNQ Architektur
- Board Setup

Bibliotheken

- ctypes

Workshop - Applikationen

- GPIO App

FPGA

- Erstellung eines Custom Overlays (Vivado)



PowerWorkshop Professional Python for Embedded

Im Bereich Embedded gilt seit langem die Programmiersprache C als der Standard. Komplexere Anwendungen und schnellere Time-to-Market-Anforderungen verlangen jedoch nach Alternativen. Python wurde traditionell zwar eher für Web und Desktop Applikationen verwendet, bietet aber dank seiner Unterstützung für C/C++ Bibliotheken und der Bereitstellung von Frameworks zur Verarbeitung von komplexen Algorithmen eine solche Alternative.

XILINX bietet ein Open-Source Framework (PYNQ), welches Python erlaubt mit dem Zynq-Portfolio zu interagieren. Dadurch können komplexere Anwendungen aus den Bereichen Analyse großer Datenmengen, maschinellem Lernen (ML) oder künstlicher Intelligenz (KI) die Vorteile der schnelleren Verarbeitung durch die programmierbare Logik ziehen. Realisiert wird dies durch hybride Bibliotheken, welche eine neue Form von Bibliotheken darstellt. Eine solche hybride Bibliothek beinhaltet sowohl einen Bitstream und den damit verbundenen hardwarenahen C-Code, aber auch eine Programmier-Schnittstelle (API) für Python. In

diesem Kurs lernen Sie die Grundlagen in der Programmierung mit Python, welche Sie in einem Workshop anwenden. Sie sind anschließend in der Lage, Prototypen auf Basis von Python auf embedded Systemen in kurzer Zeit zu entwickeln.

Als Teilnehmer des erweiterten Workshops erhalten Sie zusätzlich eine profunde Schulung in der Nutzung von Bibliotheken zur Verarbeitung großer Daten. Dies bildet die Grundlage zur Entwicklung von Anwendungen aus den Bereichen KI, ML und Big Data.

Anwendbare Technologien

Python, PYNQ

Voraussetzungen

Grundlegende Kenntnisse über objektorientierte Programmierung

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlicher Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Einführung

Erste Schritte

- Environment Setup
- Gründe für Python
- Einsatz von Jupyter

Programmierung mit Python

Grundlagen

- Variablen
- Typen, I/O und Import
- Operatoren & Namespace
- Datentypen

Abläufe

- Statements
- Loops

Funktionen

- Funktion & Argument
- Rekursion
- Modul & Package

Objekte und Klassen

- OOP
- Vererbung

Advanced

- Iteratoren
- Generatoren
- Dekoratoren
- Closures

Datenanalyse mit Python

- Strukturen & Sortierung
- Operationen & Algorithmen
- Statistiken
- "Wrangling"
- Visualisierung

Embedded Hardware

Einführung

- PYNQ Architektur
- Board Setup

Bibliotheken

- CTypes

Workshop - Applikationen

- GPIO App
- Face & Eye Detection
- HDMI Streaming & Processing

FPGA

- Erstellung eines Custom Overlays (Vivado)



Workshop Migrate to Real-Time OS: Hands-On System Development

Software development is a very dynamic process and after the initial project planning things are running smoothly in high-loop model until the first timing or scheduling problem appears. This problem exists in every other project and it could be due to a lack of detailed planning or by a continuous growth in feature set. It is now up to the development teams to fix the problem from the root or keep struggling with high-loop structure and make steady progress to meet milestones.

Code development with a High-loop structure is a single process along with some ISRs, which must pay attention to many inputs on and process in one cycle. Normally one process or IO behaves unexpectedly on a path and the whole system behaves differently. Also adding additional features in this code is a very challenging task, as a complete code must be validated once again. Sometimes this problem is due to external factors where maybe the cloud endpoint decided that the standard way of sending data is not secure and they want to implement a new standard.

It's time to think if you want continue to handle issues every day or sit down and re-plan firmware based on a pre-qualified scheduler, where each feature is handled by independent

Practical Design Experience for You!

processes/loops. Using an RTOS for a very simple project might sound like an overkill but in the long run, this decision will prove beneficial. Shifting to a reliable scheduler will give you better performance and results.

This training will introduce you to concepts for using FreeRTOS features and will show you that, how with minimum effort, a high-loop structure is adopted for RTOS.

Applicable technologies

XILINX SoC Zynq-7000 SoC and Zynq Ultrascale+ MPSoC

Requirements

Basic knowledge of embedded system programming in C

Basic knowledge of SoC or embedded processors

Duration and Costs:

Duration: 2 days

Cost: € 1,500 net per person, including detailed training material, drinks in the breaks and lunch.

Agenda

Real time operating system programming concepts for SoC:

- Understanding the FreeRTOS architecture
- How Schedulers work
- Task Management and Life Cycle
- Creating/Deleting Tasks and Task Priorities
- Working with Interrupts and task synchronization
- Working with Semaphores, Events, Mutex and Mailboxes
- FreeRTOS Memory Management Techniques

Migrate high-loop to RTOS:

- Understanding the relationship between high-loop and RTOS
- Planning FreeRTOS Initial Task
- Identify and assign correct priorities for tasks
- Dividing high-loop into multiple tasks
- Adjusting interrupts from high-loop to RTOS
- How to avoid common mistakes



Workshop Real-Time Control System Development using RTOS

Real time operating systems demand is increasing with IoT boom. Due to increase in the complexity of products and strict timing requirements, real time operating systems have become obvious choice for embedded systems. These days even the most basic embedded system solutions must provide some scheduling and real-time functionality. It is very hard to code for handling multiple IOs and processing elements in high-loop code structure. The obvious solution is to introduce a scheduler, to execute multiple real-time tasks in parallel. The scheduler provided in FreeRTOS achieves determinism by giving user the control and letting the user decide the priority for each task.

How do I get started with FreeRTOS? Why do I need FreeRTOS? How am I able to integrate FreeRTOS on a XILINX SoC?

Practical Design Experience for You!

This course will answer all questions by focusing on FreeRTOS specifically for the XILINX SoC ZYNQ 7000. Participants will be introduced to the concepts of real time multitasking, determinism, interrupts and pre-emption followed by an implementation and hands-on practice.

Applicable technologies

XILINX SoC Zynq-7000 SoC and Zynq Ultrascale+ MPSoC

Requirements

Basic knowledge of embedded system programming in C

Basic knowledge of SoC or embedded processors

Duration and Costs:

Duration: 2 days

Cost: € 1,500 net per person, including detailed training material, drinks in the breaks and lunch.

Agenda

Real time operating system programming concepts for SoC:

- Introduction to Zynq 7000
- Understanding the FreeRTOS architecture
- How Schedulers work
- Determinism and constraints
- FreeRTOS services and APIs
- Understanding pre-emptive vs non-pre-emptive schedulers

Working with FreeRTOS:

- Task Management and Life Cycle
- Idle Task and Task Priorities
- Creating/Deleting Tasks
- Working with Interrupts and task synchronization
- Inter-task Communication
- Working with Semaphores, Events, Mutex and Mailboxes
- Understand and implement Soft Timers
- FreeRTOS Memory Management Techniques

Workshop Essentials of Microprocessors

Der Workshop „Essentials of Microprocessors“ vermittelt die Grundlagen der Mikroprozessor Architektur. In dem 1-tages Kurs werden die Schlüsselemente eines allgemeinen Mikroprozessors beschrieben. Anschließend erhält der Teilnehmer eine Einführung in die Assembler Sprache. Darüberhinaus wird das Interrupt Handling, sowie das Memory Interfacing besprochen. Zusätzlich lernt der Teilnehmer die in den XILINX Technologien verwendeten MicroBlaze und ARM Cortex-A9 Prozessoren kennen. Die vom Teilnehmer durchgeführten Übungen festigen das erlernte Wissen.

Dieser Workshop vermittelt die Grundlagenkenntnisse als Vorbereitung auf die ZYNQ-7000 SoC Technologie. Als Folgekurse sind die Workshops „Compact ZYNQ-

7000 SoC for HW Designers“ und „Compact ZYNQ-7000 SoC for SW Designers“, oder der PowerWorkshop „Professional ZYNQ-7000 SoC“ zu empfehlen.

Anwendbare Technologien

Prozessor basierte Technologien

Voraussetzungen

keine

Dauer und Kosten

1 Tag, € 750,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Overview of a Generic Microprocessor

- Exercise 1: uP block design review paper exercise

Generic Assembly Language

- Exercise 2: playing computer

Lab 1: Programming the Generic Microprocessor

Generic Interrupts

Lab 2: Writing an Interrupt-Driven Program

Interfacing with Memory

Advanced Microprocessor Features

Processor Architecture Comparison

Workshop RISC-V Architecture and FPGA Implementation

Dieser Kurs führt Sie ein in die RISC-V Architektur mit einer Implementierung auf eine Xilinx FPGA Plattform.

Die Implementierung wird hier auf ein Spartan-7 FPGA Board erfolgen und die diesbezüglichen Anforderungen verdeutlicht.

Die Prinzipien der RISC-V Computerarchitektur, einer optimierten Speicherunterstützung für parallelisierte Daten- und Instruktionzugriffe werden veranschaulicht.

Auch wird ein Überblick in die RISC-V Befehlsätze vermittelt und die Grundbausteine der RISC-V-Hardware erläutert, so wie die von Interrupt Mechanismen.

Relevante Kursinhalte sind des weiteren die arithmetische Unterstützung von 32/Bit/64 Bit Festkomma- und Fließkomma-Verarbeitung bei Verwendung des RV32 bzw. RV64 Kerns wie auch die der erforderlichen Datenschnittstellen.

Hardware Aspekte wie Ressourcen, Performance, Pipelining, Speicher und Bus-Architektur werden diskutiert und in der Praxis im Vivado FPGA Projekt implementiert und analysiert.

Im Folgendem wird die Software Toolchain eingerichtet, die

für die RISC-V CPU Architektur in Ergänzung von Xilinx Bibliotheken erfolgen soll, um Xilinx-IPs wie beispielsweise Memory Controller oder GPIOs nutzen zu können.

Zum Ende des Kurses werden einfache Applikationen erstellt, um das erstellte RISC-V System auf der Xilinx Spartan-7 Plattform erfolgreich zu testen.

Der Kurs vermittelt sowohl theoretische als auch praktische Inhalte.

Anwendbare Technologien

Xilinx FPGAs

Voraussetzungen

Grundkenntnisse der HDL Programmierung

Grundkenntnisse der C/C++ Programmierung

Kenntnisse des Xilinx FPGA design flows

Dauer und kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction to Computer Architecture

- CISC, MIPS, RISC
- RISC-V Instruction Set Basic
- Memory Hierarchy

Embedded Design on XILINX FPGAs

- Xilinx FPGA Architecture Overview
- Bus Architectures: Wishbone, APB, AHB, AXI
- Embedded Design IP Core
- Hardware Implementation Flow
- Vitis and Toolchains

Computer Arithmetic in RISC-V Processor

- Integer fixed-point Arithmetic
- Floating-point Arithmetic
- Signed/Unsigned Operands
- RISC-V Repositories

RISC-V Processor Design

- Fetching the Hardware Repository
- Interrupt Architecture and Controller
- The Basic Hardware FPGA Design

- Xilinx FPGA Implementation
- Working Prototype Example

RISC-V Processor Toolchain

- RISC-V Software Repositories
- Xilinx Driver Repositories
- Setting up the Toolchain

Application Development

- Hello World Application
- Memory and Peripheral Test Applications
- Downloading and Booting

Hardware and Software Debugging

- Simulation
- Vivado Hardware Debugging
- JTAG based Processor Debugging

Die Themen begleitenden Übungen am Laptop vermitteln die praktischen Kenntnisse

Workshop RISC-V Core Verification and Compliance Testing

Dieser Kurs wurde entwickelt, um die RISC-V-Architektur, ihre Implementierung in FPGA sowie Methoden und Tools zur Überprüfung und Validierung anhand der Konformitätsspezifikation zu veranschaulichen.

Für die Validierung der RISC-V System Implementierung im FPGA wird auch eine Übersicht von Methoden und Design-Rule-Checks in Vivado vermittelt.

Die RISC-V-Computerarchitektur wird aufgezeigt, um auch anschließend ein spezifisches embedded System mit dem Core zu realisieren und die Implementierung dieser Befehlsätzen sicherzustellen.

Der Überblick über die Grundbausteine der RISC-V-Hardware mit Erläuterung von Methoden und Werkzeuge ermöglicht umfassende Überprüfungen von Funktionalität in Simulation und Hardware Ausführung.

Auch die arithmetische ALU Unterstützung wie Festkomma- und Fließkommaverarbeitung wird mit veranschaulicht.

Der Prozess des Aufbaus einer Verifizierungsumgebung unter Verwendung verschiedener Ansätze wird gezeigt.

Der Kurs beinhaltet die Implementierung mit einem RV32/RV64 compliant RISC-V CPU-Kern.

Die Entwicklung der Software wird dargestellt, um das eingebettete System auf der Xilinx FPGA-Plattform zu realisieren.

In der zu definierenden Testumgebung muss die Funktionalität des erstellten embedded Projektes simuliert werden können.

Behandelt werden Methoden der FPGA Implementierungs-Analyse um Aussagen von Ressourcen, Echtzeitfähigkeit und Zuverlässigkeit sicherzustellen.

Der Kurs vermittelt sowohl theoretische als auch praktische Inhalte.

Anwendbare Technologien

Xilinx FPGAs

Voraussetzungen

Grundkenntnisse von Linux Shell Scripting
Grundkenntnisse der C/C++ Programmierung
Kenntnisse des Xilinx FPGA design flows

Dauer und kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

RISC-V Instruction Set Basics

- Instruction Set Principles
- Memory Addressing
- Representation of Instruction Operations
- Representation of Operands

Xilinx FPGA Implementation Analysis

- UltraFast Design Methodology
- HDL Coding Techniques
- Synchronization Techniques
- Clocking and Timing Analysis
- Creating various Vivado Reports

Introduction to Processor Verification

- Verification Environment
- Verification Methodology
- Golden Model Development
- Verification Environment for Core Simulation
- Requirements for Hardware Debugging

RISC-V Compliance Definition

- Compliance Test Framework
- Compliance Test Pool
- Test Suite Development for checking CPU Instructions

RV32 Core Simulation and Compliance Tests

- Core Structure
- Simulation Environment
- Tests pool
- Compliance Testing and Results Analysis
- RV32 Core Demonstration and In-Circuit Debugging

Übungen

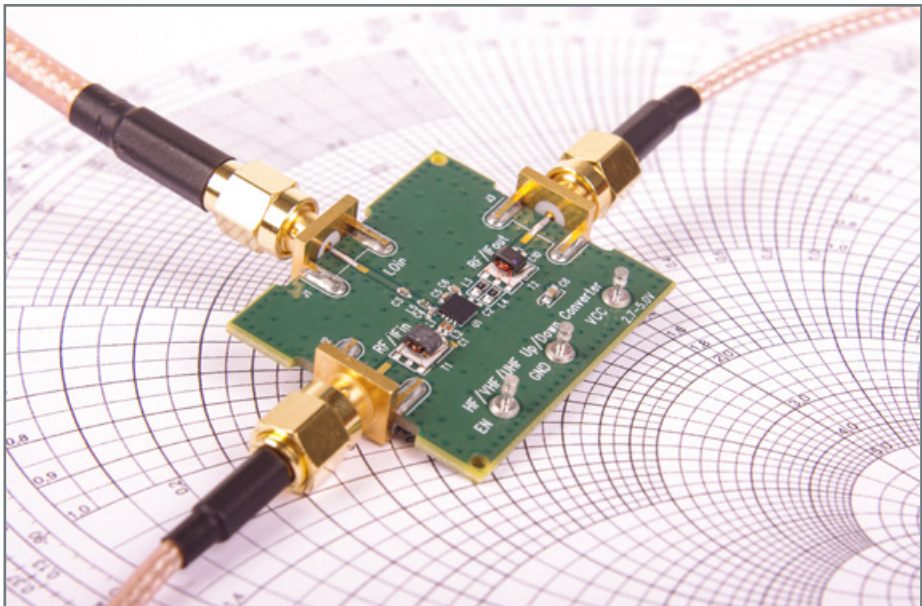
- **Labs:** Umfangreiche themenbegleitende Übungen ermöglichen eine praxisgerechte Einarbeitung

Ausbildungsschwerpunkt „DSP & Image Processing“

Die in diesem Bereich zusammengefassten Workshops beinhalten die für Sie wichtigen Informationen um DSP Anwendungen mit XILINX FPGAs entwickeln zu können. Das Angebot um-

fasst sowohl die Verwendung des XILINX System Generators im Zusammenspiel mit Matlab Simulink als auch die HDL basierende Entwicklung von DSP Funktionen.

DSP & Image Processing	Seite	Dauer (Tage)	Kosten
DSP Design using System Generator	98	3	€ 2.100 o. 27 TCs
Model Composer	99	2	€ 1.500 o. 18 TCs
Expert DSP Design	100	5	€ 3.100 o. 45 TCs



Workshop DSP Design using System Generator

Mit der Vivado ML Editionen wurde die Vivado Tool Option Xilinx System Generator mit Model Composer verschmolzen, das dem Entwickler nun verschiedene Flow Methoden ermöglicht. Dieser Workshop fokussiert die Arbeitsweise auf die Toolbox System Generator, welches die HDL Code-Generierung ermöglicht. So werden die erforderlichen Kenntnisse vermittelt, DSP Strukturen in grafischer Umgebung zu entwickeln, zu simulieren und im Xilinx FPGA zu implementieren. Insbesondere Möglichkeiten, die FPGA Hardware an das Simulink Werkzeug zu koppeln, sowie umfangreiche Stimulus Elemente der Matlab Libraries verkürzen den Design Zyklus der HDL Verifikation. Ein weiterer Vorteil besteht in der Methodik, nichtlineare Randbedingungen wie Wortbreite, Sättigungs- und Rundungsverhalten vor der Implementierung zu analysieren.

Vorkenntnisse zu Grundlagen digitaler Signalverarbeitung werden vorausgesetzt.

Anwendbare Technologien

Xilinx FPGAs, Zynq SoCs and MPSoCs, Versal ACAP

Voraussetzungen

Grundkenntnisse der digitalen Signalverarbeitung

Von Vorteil: Grundkenntnisse Xilinx FPGA Technologie

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer
inklusive ausführlichen Schulungsunterlagen
sowie Pausengetränken und Mittagessen

Agenda

Matlab Simulink Basics

- Stimulus and Response
- Sampling and Resolution
- Design Hierarchy and Subsystems

Xilinx System-Generator Basics

- Interfacing and Data Types
- Constructing the Design
- System-Generator Blocks

Digital Filter Implementations

- FIR Filter Compiler
- Parallel Filters
- Sequential Filters

Verification

- Simulink Verification
- HDL Co-Simulation
- Hardware Verification

Signals in System-Generator

- Signal Conversion
- Signal Extraction

Implementing System Control

- Control Mechanisms
- Block Operations
- Implementing M-File Functions

Model Composer

- Model Composer Introduction
- Model Composer Optimized Block Library
- Compilation and Simulation

Vivado IDE Integration

- System Generator Integration

Übungen

- Using the Simulink Software
- Getting Started with Xilinx System Generator
- Signal Routing
- Implementing System Control
- Designing a FIR Filter Designs
- Model Composer Video Processing
- DSP Module Integration in the Vivado IDE

Workshop Compact Model Composer

Der 2-tägige Workshop bietet dem System- und Algorithmenentwickler mit dem Xilinx Tool Model Composer eine Einarbeitung in eine Methodik auf höher abstrahiertem Level Hardware Code automatisch zu generieren.

Mit Vivado ML Edition sind in der Model Composer Toolbox nun auch Xilinx System Generator und HLS Tool integriert:

Die HDL Toolbox basiert auf Xilinx System Generator, die HLS Toolbox basiert auf Xilinx Vitis und AI Engine unterstützt die Code Generierung auf AI Engines mit Versal Technologien.

Unter Matlab Simulink werden auf diese Weise DSP Module entwickelt, die modular in das FPGA/SoC/MP-SoC/ACAP Projekt eingebunden werden.

Mit dem Xilinx Tool Model Composer können komplexe Algorithmen sehr schnell Code-generiert werden, ohne Dateilwissen von FPGA Technologie oder VHDL/Verilog Programmiersprachenkenntnisse.

Entwickler, die Aufgaben im Kontext von digitaler Modellierung, Videoprocessing, Algorithmen umsetzen, nutzen mit diesem Werkzeug eine höhere Methodenabstraktion als mit dem Tool Xilinx System Generator,

hoch effiziente Optimierungsergebnisse sowie einfache Validierung unter Matlab Simulink.

Das Einbinden eigener C/C++ Templates wird unterstützt sowie manigfaltige Schnittstellen zu FPGA Modulen wie Xilinx HLS oder VHDL/Verilog.

Dieser Workshop empfiehlt sich für Teilnehmer, die mit Matlab Simulink auf die Zielplattformen von Xilinx Technologien arbeiten.

Anwendbare Technologien

Xilinx FPGAs, Zynq SoCs und MPSoCs, Versal ACAP

Voraussetzungen

Grundkenntnisse Digitaler Signalverarbeitung

Grundkenntnisse in C/C++

Grundkenntnisse von Matlab/Simulink sind von Vorteil

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Introduction to Model Composer
- Matlab Simulink Overview
- Model Composer Block Library
- Data Types Support
- Creating a Model Composer Design
- Importing C/C++ Code as Custom Blocks
- Simulink based Simulation
- Integration of Model Composer Projects
- Model Composer - Using the HDL Toolbox
- Model Composer - Using the HLS Toolbox
- Model Composer - Using the AIE Toolbox
- Model Composer to Vivado IPI
- Processor Acceleration using Zynq Plattformen
- Hardware Co-Simulation

Labs

- Introduction to Model Composer
- Introduction to System Generator
- Importing C/C++ templates
- Model Composer Graph Flow for Versal
- Model Composer Module Integration
- Hardware Co-Simulation

PowerWorkshop

Expert DSP Design

Der Workshop „Expert DSP Design“ vermittelt die entscheidenden Entwicklungskennnisse für DSP und Algorithmenentwicklung für die Hardware-Implementierung für Xilinx Technologien. Grundlage dieses Workshops ist die Arbeitsweise mit Matlab Simulink. Es werden unterschiedliche Entwicklungsmethoden erläutert, so wie Design Flow, Simulation/Co-Simulation, Optimierung, Analyse, Implementierung und auch die Integration in die Zielplattform.

Beginnend mit Xilinx System Generator wird insbesondere die Hardware-nahe Programmierung mit dieser Methodik bevorzugt. Eine höher abstrahierte Entwicklungsmethodik ist mit der Toolbox HLS zu arbeiten, die insbesondere auch für Video Applikationen präferiert ist.

Die dritte Methodik, die aufgezeigt wird, ist Vitis HLS, wo insbesondere kundenspezifische Algorithmen entwickelt werden, die als solche mit den Toolboxes nicht zur Verfügung gestellt werden.

Die klassische VHDL/Verilog Entwicklungsmethodik wird hier nicht angewandt, wohl aber wird gezeigt, wie solche Module eingebunden werden können und auch Co-simuliert werden. Die Integration in die Zielplattform (FPGA, Zynq SoC/MPSoC oder Versal ACAP) schließt diesen Workshop ab, wobei DSP Module mit den aufgezeigten Methoden einzubinden sind.

Anwendbare Technologien

Xilinx Zynq UltraScale+ MPSoC

Voraussetzungen

Grundlagenkenntnisse Digitale Signalverarbeitung
Programmierkenntnisse C/C++

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer
inklusive ausführlichen Schulungsunterlagen
sowie Pausengetränken und Mittagessen

Agenda

Matlab Simulink Basics

- Sampling and Resolution
- Design Hierarchy and Subsystems

Model Composer

- Model Composer Introduction

Xilinx System-Generator Basics

- Interfacing and Data Types
- System-Generator HDL Blockset

Digital Filter Implementations

- Sequential and Parallel Filters
- FIR Filter Compiler

Verification

- Simulink Verification
- HDL and Hardware Co-Simulation

Signals in System-Generator

- Signal Conversion
- Signal Extraction

Implementing System Control

- Control Mechanisms
- Implementing M-File Functions

Model Composer

- Model Composer HLS Block Library
- Compilation and Simulation

Vivado IDE Integration

- Model Composer Modules Integration
- HLS Module and HDL Module Integration

Übungen

Umfangreiche themenbegleitende Übungen ermöglichen eine praxisperechte Einarbeitung

Ausbildungsschwerpunkt „Connectivity“

Die in diesem Bereich zusammengefassten Workshops beinhalten detaillierte Informationen über die Realisierung von Kommunikationsprotokollen wie Ethernet oder PCI Express sowie die Anbindung von externen Speichern. Bei diesen High-Speed Protokollen ist die Signalintegrität sowie

die Einhaltung von Grundprinzipien beim FPGA Board Design besonders wichtig.

Sie können sich jederzeit von unseren erfahrenen Trainern und Referenten im Vorfeld beraten lassen.

Connectivity	Seite	Dauer (Tage)	Kosten
High-Speed Memory Interfacing	103	3	€ 2.100 o. 27 TCs
DDR4 Interfacing with Xilinx FPGAs	104	3	€ 2.100 o. 27 TCs
Designing with Multi-Gigabit Serial I/O	105	3	€ 2.100 o. 27 TCs
Designing with Ethernet MAC Controllers	106	2	€ 1.500 o. 18 TCs
Designing with PCI Express	107	3	€ 2.100 o. 27 TCs
Professional PCI Express	108	5	€ 3.100 o. 45 TCs
Signal Integrity	109	3	€ 2.100 o. 27 TCs
FPGA Board Design	110	2	€ 1.500 o. 18 TCs
ZYNQ Board Design & High-Speed Interfacing	111	3	€ 2.100 o. 27 TCs
UltraScale Connectivity	112	5	€ 3.100 o. 45 TCs
VERSAL ACAP Connectivity	113	5	€ 3.100 o. 45 TCs

Workshop

High-Speed Memory Interfacing

Die neuen XILINX FPGA Plattformen gestatten schnelle Interfaces zu externen Speichern. Dabei kommen neue IOBs in den FPGAs und weiterentwickelte Wizards zum Generieren der Controller zum Einsatz. So verlagern sich die Herausforderungen in der Realisierung vom FPGA auf das PCB. Bei den hohen Datenraten bedarf das Boarddesign sehr hohe Aufmerksamkeit.

Der 3-tägige PLC2-Workshop „High-Speed Memory Interfacing“ richtet sich sowohl an FPGA Entwickler als auch an System- und Layoutdesigner, die schnelle Memoryschnittstellen implementieren und im System nutzen wollen.

Mit diesem Workshop wird der Entwickler in die Lage versetzt, die verfügbaren Optionen der IOBs effektiv einzusetzen und diese gemeinsam mit dem PCB-Design erfolgreich für das Memory-Interfacing einzusetzen. Dabei stehen die Belange der Signalintegrität im Vordergrund. Es werden

sowohl die Signalqualität als auch das Timing eingehend behandelt. Anhand von Boards mit DDR3 Memories werden praktische Design- und Verifikationsbeispiele erläutert. Die Simulationsoptionen werden vorgestellt. Im Kurs werden Richtlinien für ein SI-gerechtes Layout schneller Memory-Interfaces erarbeitet.

Anwendbare Technologien

7 Series FPGAs, bei Bedarf auch Spartan-6 und Virtex-6 FPGAs

Voraussetzungen

Grundlegende Kenntnisse über VHDL und FPGA-Implementierung

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction

- FPGA technology overview
- Memory overview (DDRx, LPDDRx, QDR, RDRAM)
- DDR3 specific features (write / read leveling, dyn. ODT, reset, ...)
- FPGA resources for memory interfaces

Logical Design

- Usage of the XILINX resources
- Design options and challenges
- Design parameters
- External memories in embedded designs

Physical Design

- PCB design parameters
- PCB design requirements
- Board-Level simulations
- PCB design rules
- Power supply

System Analysis

- FPGA timing analysis
- PCB timing analysis
- System timing analysis
- Design parameter optimization
- Summarizing all design rules

Test and Debugging

- Functional verification
- Option and strategies for debugging

Labs:

- DDR3 controller design
- Simulation memory interface
- Implementation memory interface
- Test and debugging DDR3 memory interface
- QDR II+ Interface
- MIG in embedded design - IPI
- Verification physical layer (reflection and crosstalk)
- Verification physical layer (timing)
- Post-Layout analysis

Workshop DDR4 Interfacing with XILINX FPGAs

Die neuen XILINX Architekturen gestatten sehr schnelle Interfaces zu externen Speichern. Dabei kommen weiterentwickelte Strukturen auf den Devices und ein neuer Wizard zum Generieren der Controller zum Einsatz.

Mit den neuen UltraScale Architekturen verlagern sich verstärkt Realisierungsherausforderungen vom Chiplevel hin zum Boardlevel. Die sehr hohen Datenraten erfordern extreme Sorgfalt im PCB-Design.

Der 3-tägige Workshop "DDR4 Interfacing with XILINX FPGAs" richtet sich sowohl an Hardwaredesigner als auch an Systemarchitekten und Layoutdesigner, die DDR4 Interfaces erfolgreich im System einsetzen wollen.

Mit diesem Workshop wird der Entwickler in die Lage versetzt, die verfügbaren Optionen der IOBs und Taktressourcen effektiv einzusetzen. Dabei stehen auch die Belange der Signalintegrität im Vordergrund. Es werden sowohl die Parameter der Signalqualität als auch das Timing eingehend behandelt. Anhand eines Boards mit DDR4

Memories werden praktische Design- und Verifikationsbeispiele erläutert. Die PCB-Level Simulationsoptionen werden ausführlich vorgestellt und mittels Übungen demonstriert. Im Kurs werden Richtlinien für eine erfolgreiche Realisierung von schnellen Memory-Interfaces erarbeitet. Methodische Hinweise zur erfolgreichen DDR4 Realisierung runden den Workshop ab.

Anwendbare Technologien

Alle UltraScale FPGA / MPSoC Familien und Versal ACAP

Voraussetzungen

Grundlegende Kenntnisse über VHDL und FPGA-Implementierung

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction DDR4 SDRAMs

- Enhancements over DDR3
- DDR4 Specific Features
- Operational Details

Logical Design

- Usage of the XILINX resources
- Options and challenges
- Design parameters

System Analysis

- FPGA timing analysis
- PCB timing analysis
- System timing analysis
- Design parameter optimization
- Summarizing all design rules

DDR4 Signal Integrity Compliant Design

- Simulation options
- Transmission lines
- Trace parasitics (vias, etc.)

- Pre-Layout simulation
- Post-Layout simulation
- Derivation of PCB design rules Test and Debugging
- Functional verification
- Options and strategies for logical debugging
- Options and strategies for physical debugging

DDR4 Design Methodology

- Summary

Übungen

- Controller design
- Memory interface simulation
- Controller implementation
- Debugging memory interfaces
- DDR4 pre-layout simulations
- DDR4 post-layout simulations
- Memory interfaces in IP Integrator

Workshop

Designing with Multi-Gigabit Serial I/O

Die neuen XILINX FPGA Plattformen gestatten serielle Interfaces bis in den 58 Gbit/s-Bereich. Dabei kommen neue Transceiver zum Einsatz, für deren schnelle und effektive Nutzung ein umfangreiches Wissen erforderlich ist.

Der 3-tägige PLC2-Workshop „Designing with Multi-Gigabit Serial I/O“ richtet sich an Entwickler, die serielle Schnittstellen im Gbit/s-Bereich implementieren und im System nutzen wollen.

Mit diesem Workshop wird der Entwickler in die Lage versetzt, alle verfügbaren Features der Transceiver effektiv einzusetzen, die für die Features notwendigen Schnittstellen und Attribute festzulegen, mit dem im IP Catalog integrierten Transceiver Wizard die seriellen Transceiver Primitives zu instanzieren, um somit projektspezifische Schnittstellen oder standardisierte serielle Interfaces zu

realisieren. Mit dem IBERT Design wird eine effektive Verifikation der seriellen Übertragungstrecke gezeigt. Themen zur Signalintegrität einschließlich eines Simulationsbeispiels und Hinweise zum Boarddesign geben praktische Hinweise zur Realisierung.

Anwendbare Technologien

Aktuelle FPGA-Technologien

Voraussetzungen

Grundlegende Kenntnisse über VHDL und FPGA-Implementierung

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction

- FPGA technology overview
- GTP/GTX overview
- GTH/GTY overview
- GTM overview

Clocking, timing and RESET

Physical Coding Sublayer

- Fabric interface
- Encoding / decoding
- Symbol alignment, clock correction
- Channel bonding

Physical Media Attachment

- Serial IO
- Pre-Emphasis and equalization

Implementation and Verification

- Architecture Wizard
- SecureIP simulation
- IBERT design

System Design

- 64B/66B and 64B/67B encoding and the Gearbox
- PMA Layer details
- Boarddesign
- Link optimization
- Test and debugging

Labs:

- Coding/encoding lab
- Using Commas and K-character symbols
- Clock correction
- Channel bonding
- Using Architecture Wizard
- Implementation and Simulation
- 64B/66B encoding
- System-Lab: from Specification to Debugging / Link Optimization

Workshop

Designing with Ethernet MAC Controllers

Das Ethernet Protokoll findet in vielfältigen Applikationen Anwendung. XILINX offeriert hierfür verschiedene Soft- und embedded Hardware Realisierungen, für deren schnelle und effektive Nutzung ein umfangreiches Wissen erforderlich ist.

Der 2-tägige PLC2-Workshop „Designing with Ethernet MAC Controllers“ schult den Designer im praktischen Einsatz von Ethernet Implementierungen im FPGA. Dabei wird, ausgehend von den Grundlagen des Ethernet Standards auf Protokollbeschreibung und OSI Schichtenmodell eingegangen. Die Anwendung der XILINX spezifischen Ethernet Lösungen wird beschrieben und anhand von Übungen gefestigt.

Dem Teilnehmer wird das für die erfolgreiche Implementierung notwendiges Wissen zum Hardwaredesign sowie zu der Softwareentwicklung vermittelt. Zum Abschluss

des Workshops wird ein Ethernet basiertes Beispiel in realer Hardware auf einem XILINX FPGA Evaluation Board implementiert und getestet.

Die Übungen schließen Ethernet-Hardware Implementierungen mit einem Kintex-7 Evaluierungsboard ein.

Anwendbare Technologien

XILINX FPGA, SoCs, MPSoCs, RFSocS und ACAPs

Voraussetzungen

Gute Kenntnisse der XILINX FPGA Architekturen
Grundlegende VHDL Kenntnisse.

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

- Ethernet Basics
- Network Protocols
- Ethernet Frames
- MAC: Media Access Control
- Network Protocols
- Ethernet Hardware
- Physical Layer
- Ethernet Interfaces
- Flow Control
- Management Interface
- XILINX EMAC Solutions
- 10/100 EMAC Solutions
- TEMAC
- 10GE/25GE/40GE MAC
- Integrated IP: 100G Ethernet MACs

Übungen

- **Lab:** Analyzing Ethernet Frames
- **Lab:** VLAN and Jumbo Frames
- **Lab:** Implementation
- **Lab:** EMAC Peripheral in Loopback Mode
- **Lab:** TEMAC Peripheral in Loopback Mode
- **Lab:** Analyzing 10GE/25GE/40GE MAC Frames

Workshop Designing with PCI Express

Mit der Verfügbarkeit der neuen FPGA-Familien ergeben sich effektive PCI Express (PCIe) Lösungen, besonders unter Verwendung der auf den Chips vorhandenen IP-Blöcke. Zu Beginn gibt dieser Kurs eine Einführung in die Nomenklatur, das Übertragungsprotokoll und wichtige Details zur Erzeugung von Datenpaketen. Damit verfügt der Teilnehmer über eine solide Grundlage für die Implementierung seiner Applikation. Der Schwerpunkt des Kurses liegt dann auf der Simulation und Umsetzung der Aufgabenstellung an einem konkreten Beispiel. Damit wird ein bestmöglicher Realitätsbezug geschaffen und eine effiziente Wiederverwendbarkeit des Erlernten gesichert. Besonderes Augenmerk ist auf das User-Interface gelegt, welches die Schnittstelle zwischen der Userapplikation und dem PCIe-Block selber darstellt. Die vorhandenen Signale und deren korrekte Verwendung werden detailliert besprochen.

Während des Kurses wird mit Kintex-7 / Kintex UltraScale als Zielhardware gearbeitet. Das Erlernte ist jedoch prinzipiell auf die anderen Technologien übertragbar.

Anwendbare Technologien

XILINX 7 Series FPGAs, alle UltraScale Architekturen

Voraussetzungen

Grundkenntnisse VHDL

Grundkenntnisse im XILINX Design Tool Flow

Grundkenntnisse FPGA-Technologie

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction to the PCIe Architecture

- Overview and Architecture
- Protocol Basics
- Performance

Review of the PCIe Protocol

- Transaction Types and Categories
- Virtual Channels and Flow Control
- Address and Configuration Space

PCIe and the IP Catalog Interface

- Core Selection
- Configuration Space

Simulating a PCIe System Design

- Identifying Simulation Points
- Simulation Methods
- Building Testbenches

Connecting Logic to the Core – AXI Interface

- PCIe Link and System Interface Signals
- Common Transaction Interface Signals
- Transmit Interface Signals
- Receive Interface Signals
- Configuration Space Interface Signals
- Physical Layer Interface Signals

Packet Formatting Details

- General TLP Format
- MemW/MemR
- Cpl/CpID
- Msg/MsgD

Endpoint Application Considerations

- Design Specification and Considerations
- Selecting the Appropriate Core
- Specific Register Awareness
- Endpoint Responsibilities

Application Focus DMA

Root Port

PCIe Configuration

- Tandem Configuration
- Software Flow Details

Compliance and Debugging

- Logic Analyzer Tool and Debugging a PCIe Core Endpoint
- Compliance Testing

Interrupts and Error Management

- PCIe Core Errors
- Interrupts

PowerWorkshop Professional PCI Express

Mit der Verfügbarkeit der neuen FPGA-Familien ergeben sich effektive PCI Express (PCIe) Lösungen, besonders unter Verwendung der auf den Chips vorhandenen IP Blöcke.

Zu Beginn gibt dieser Kurs eine Einführung in die Nomenklatur, das Übertragungsprotokoll und wichtige Details zur Erzeugung von Datenpaketen. Damit verfügt der Teilnehmer über eine solide Grundlage für die Implementierung seiner Applikation. Besonderes Augenmerk ist auf das User-Interface gelegt, welches die Schnittstelle zwischen der User-Applikation und dem PCIe Block darstellt.

Die vorhandenen Signale und deren korrekte Verwendung werden detailliert besprochen. Während des Kurses wird mit Kintex UltraScale als Zielhardware gearbeitet. Das Erlernete ist jedoch prinzipiell auf die anderen Technologien übertragbar.

Dieser Kurs bietet detaillierte Protokollinformationen und wertvolle Hinweise zum PCB-Design und Details zum PCIe Hard-IP. Das erlernte theoretische Wissen wird durch

einen intensiven Praxisteil vertieft. Nachdem die wesentlichen Punkte durch Simulationen erläutert wurden, wird ausführlich auf Softwareaspekte eingegangen. Nach allgemeinen Software und Treiber Diskussionen wird in dem Praxisteil auf DMA Konzepte und andere Applikationen näher eingegangen. Damit wird das praktische Verständnis geschaffen, um direkt die XILINX IP Cores in dem eigenen Design verwenden zu können.

Anwendbare Technologien

Alle Ultrascale Architekturen und Versal ACAP

Voraussetzungen

Grundkenntnisse VHDL

Grundkenntnisse XILINX Design Tools

Grundkenntnisse FPGA-Technologie

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction to the PCIe Architecture

PCIe Protocol Specification

- PCIe Layers and Channels
- TLP Fields and Packet Routing
- Requests and Completion Packets
- PCIe Configuration Space
- Errors, Interrupts and Performance

XILINX PCI Express Solutions

PCIe Core Customization

Simulating a PCIe System Design

- Identifying Simulation Points
- Simulation Methods
- Building Testbenches

Connecting Logic to the Core

- Introduction to AXI
- PCIe IP Interfaces

Endpoint Application Considerations

- Design Specification and Considerations
- Endpoint Responsibilities
- Interpreting Data from the Core

Application Focus DMA

Root Port Design

Zynq UltraScale+ PS PCIe Controller

PCIe Configuration

- Tandem Configuration
- Software Flow Details

Compliance and Debugging

- Debugging a PCIe Core
- Compliance Testing
- Link Training Debugging

Mechanicals, Hot Plug, and Power

PCIe Board Design

Device Driver Basics

- Driver Model Overview
- Tool-flows for Linux, PetaLinux and Windows

Device Driver Practice

- Tracing and Debugging
- DMA Concepts, Scatter/Gather Transfers
- Notifications : Interrupts and Doorbells
- Managing Circular Buffers
- Synchronous and Asynchronous I/O
- Caching Issues

Labs

Workshop Signal Integrity

Die neuen XILINX FPGA Familien gestatten Interfaces mit hohen Takt- bzw. Datenraten, bei denen der Entwickler gezwungen ist, die Effekte der Signalintegrität zu berücksichtigen.

Der 3-tägige PLC2-Workshop „Signal Integrity“ richtet sich an Entwickler, die high-speed Schnittstellen zwischen einem XILINX FPGA und anderen Komponenten implementieren wollen. Der Teilnehmer wird in die Lage versetzt, zu beurteilen, wann die Signalintegrität zu berücksichtigen ist. Zur weiteren Analyse stehen z.B. IBIS Modelle zur Verfügung, deren Einsatz und Erstellung beschrieben werden. Diese Modelle bilden die Grundlage für weitergehende Analysen (Reflexionen, Übersprechen, etc.) mit HyperLynx. Basierend auf diesen grundlegenden Analysen lernt der Teilnehmer die Simulation schneller paralleler In-

terfaces, wie beispielsweise DDR3 / DDR4 Memories. Die Simulation der seriellen Schnittstellen erfolgt basierend auf verschiedenen Designkits, deren designspezifische Anpassung mit einer Lab gezeigt wird.

Anwendbare Technologien

Aktuelle FPGA/SoC/MPSoC-Architekturen

Voraussetzungen

Grundkenntnisse Schaltungstechnik und Umgang mit Windows-Programmen

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction and Architecture

- FPGA technologies overview
- Spartan technologies overview
- Clocking resources
- SelectIO and serial interfaces
- Packaging
- Configuration overview

Transmission Lines

- Basics
- Critical trace length

IBIS Models and Signal Integrity Tools

- IBIS standard
- IBIS editor
- XILINX IBIS models
- SI tools
- HyperLynx

Reflections and Crosstalk

- Reflection effects and calculation
- Minimizing reflections: trace termination
- Crosstalk effects and calculations
- Minimizing crosstalk

Signal Integrity Analysis

- Methods in general
- Modeling of components
- System analysis:
 - High-speed memory interfaces
 - Serial transceiver

Power Supply

- Impedance – inductance
- Bypassing

Technology Specific Solutions

- Powering
- Signal interfacing
- Packaging and thermal aspects

Labs:

- Invoking HyperLynx
- Analysis of reflection effects
- Analysis of crosstalk effects
- Memory interface example
- Serial I/O example

Workshop FPGA Board-Design

Mit der Verfügbarkeit der verschiedenen XILINX FPGA Familien werden sowohl die FPGA Designer als auch die Layoutentwickler mit neuen Realisierungsmöglichkeiten und Herausforderungen konfrontiert. Verschiedenste Schnittstellen zu externen Komponenten ermöglichen vielfältige Einsatzgebiete. Die dabei möglichen hohen Taktraten erfordern ein umfangreiches Wissen über das physikalische Design im Umfeld des FPGA.

Der 2-tägige PLC2-Workshop „FPGA Board Design“ schult die Designer im praktischen Einsatz von XILINX FPGAs. Der Workshop richtet sich nicht nur an VHDL Entwickler, die auch physikalische Schnittstellen des FPGA einzubeziehen haben, sondern auch an Konstrukteure und PCB-Layer. Dabei wird auf die sich zum Teil widersprechenden Designrestriktionen beider Designteams eingegangen, um somit Methoden zur effektiven Realisierung der Kundenprojekte zu übermitteln.

Basierend auf den Power-Anforderungen des FPGA (Grundlagen und Power-Estimator) werden Lösungen für die Stromversorgung (AC/DC bzw. DC/DC Wand-

ler, Filtering und Decoupling) vorgestellt. Auf die für hohe Takt- bzw. Datenraten erforderliche Terminierung wird umfassend eingegangen, dabei werden verschiedene Signalpegel und Terminierungsvarianten berücksichtigt. Ein spezielles Kapitel behandelt die Taktversorgung (Strategien und Realisierung) und die Verbindung zu high-speed Komponenten auf dem Board. Weiterhin werden Regeln zum Leiterplattendesign (PCB tracing, Layer Stacking) erläutert.

Anwendbare Technologien

Aktuelle FPGA-Technologien

Voraussetzungen

Grundkenntnisse Schaltungstechnik und Umgang mit Windows-Programmen

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction and Architecture

- FPGA technologies overview
- Clocking resources
- SelectIO and serial interfaces
- Packaging
- Configuration overview

FPGA Power Supply

- Power estimation
- Power supply solutions
- Power filtering and decoupling

FPGA High-Speed Interfaces

- High-speed signal standards
- Signal interfacing
- Transmission lines
- Combining I/O standards
- Signal coupling and translation
- SSO
- DCI / OCT

FPGA Clocking

- High-speed clock generation
- Clock distribution
- Board de-skewing

Physical Design, Modeling and Simulation

- PCB structures
- Trace characteristics
- Board design rules
- PCB modeling and simulation

Thermal Aspects

- Thermal resistance, modeling and calculation, heat sink selection

Labs:

- Multiple Labs on
 - I/O placement with PlanAhead
 - Power prediction
 - Demo reflection and crosstalk using HyperLynx
 - Thermal design

Mit den neuen ZYNQ Chips verlagern sich auch Realisierungsherausforderungen vom Chiplevel hin zum Boardlevel. Die sehr hohen Datenraten erfordern sehr viel Sorgfalt im PCB-Design.

Der 3-tägige Workshop “ZYNQ – Board Design and High-Speed Interfacing” richtet sich sowohl an Hardwaredesigner als auch an Systemarchitekten und Layoutdesigner, die ZYNQ Bausteine erfolgreich im System einsetzen und schnelle Interfaces realisieren wollen.

Dieser Workshop startet mit einer Diskussion der Signal- und Powerintegrität. Mit dem Modul PCB Interface Optionen und Bedingungen wird die Notwendigkeit der Berücksichtigung der einleitenden theoretischen Grundlagen deutlich. Im nachfolgenden werden für Stromversorgung und „normales (– low-speed)“ Interfacing entscheidende Details diskutiert. Auch für das Design mit dem on-chip A/D-Wandler werden wichtige Hinweise gegeben. Schwerpunkt liegt auf einer detaillierten Be-

schreibung der schnellen Interfaces für Transceiver und Memory Anschlüsse. Hier sind teilweise sehr spezielle Designregeln einzuhalten. Wie für die Stromversorgung werden auch hier praxisnahe Beispiele besprochen und Anregungen für das Kundendesign vermittelt. Betrachtungen zum thermischen Design runden den Workshop ab. Abschließend werden Hinweise zum Troubleshooting auf PCB-Ebene behandelt.

Anwendbare Technologien

ZYNQ-7000 / ZYNQ UltraScale+ MPSoC/RFSoc

Voraussetzungen

Grundlegende Kenntnisse über ZYNQ Hardware Architektur sind wünschenswert

Dauer und Kosten

3 Tage, € 2.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Overview Board Level Interface Design Challenges

Basics Signal Integrity

- Effect of fast signal edges
- Transmission lines
- Reflection and termination
- Simulation options

PCB Interface Options and Requirements

- Pins and packaging
- Power requirements
- Signal interfacing requirements

Power Management

- Power estimation and tools
- Power distribution
- Filtering and decoupling
- Powering and sequencing realization options

Parallel Interfacing (PS and PL Sub-systems)

- I/O Standards and termination
- PCB rules and pin definition

XADC – Mixed Signal

- Signal conversion and noise
- PCB rules

Serial Transceiver

- Transceiver overview
- Powering and signal interfacing
- PMA layer options
- PCB guidelines
- Physical link setup and optimization
- PCI Express specific rules

High-Speed Memory Interfaces

- Hard IP vs. Soft IP
- Physical interface realization
- Options for SI-simulation
- Physical interface verification and debugging

Thermal Design

- Thermal resistance, modeling and calculation
- Heat sink selection

PCB-Level Troubleshooting

PowerWorkshop UltraScale FPGAs – Connectivity

Die XILINX UltraScale Architekturen gestatten sehr schnelle Interfaces zu externen Komponenten. Auf den Bausteinen existieren weiterentwickelte Strukturen für die neue Wizards zur Konfiguration in der Entwicklungsumgebung zur Verfügung stehen.

Mit den neuen UltraScale Architekturen verlagern sich verstärkt Realisierungsherausforderungen vom Chiplevel hin zum Boardlevel. Die sehr hohen Datenraten erfordern extreme Sorgfalt im PCB-Design.

Der 5-tägige Power-Workshop „UltraScale FPGAs - Connectivity“ richtet sich sowohl an Hardwaredesigner als auch an Systemarchitekten und Layoutdesigner, die schnelle Interfaces erfolgreich im System einsetzen wollen. Mit diesem Workshop wird der Entwickler in die Lage versetzt, die verfügbaren Optionen der IPs und Taktressourcen effektiv einzusetzen. Schwerpunkt sind DDR4, Transceiver und PCI Express Interfaces. Dabei stehen neben IP Core Konfiguration, Simulation und Implementie-

rung auch die Belange des PCB Designs und der Signalintegrität im Vordergrund. Anhand eines UltraScale FPGA Boards werden praktische Design- und Verifikationsbeispiele erläutert. Die PCB-Level Simulationsoptionen werden vorgestellt. Im Kurs werden Richtlinien für eine erfolgreiche Realisierung von schnellen Interfaces erarbeitet. Methodische Hinweise zur erfolgreichen Realisierung schneller Interfaces runden den Workshop ab.

Anwendbare Technologien

UltraScale Architekturen

Voraussetzungen

Grundlegende Kenntnisse über VHDL und FPGA-Implementierung

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Agenda

Introduction

Architecture Overview and Clocking

- UltraScale FPGAs architecture
- IO resources
- Clock resources

Signal Interfacing

- IO standards
- Termination
- Banking rules
- Data Converter
- PCB rules and pin definition

Configuration

DDR4 Memory Interfacing

- Enhancements DDR4 over DDR3
- DDR4 specific features
- Controller architecture and interfaces
- Configuring DDR4 controller
- Memory interface simulation and implementation
- Debugging DDR4 interfaces
- PCB requirements and design
- Signal Integrity simulation options

Serial Transceiver Interfacing

- Transceiver overview
- PCS blocks and usage options
- PMA layer options
- Simulation and implementation
- PCB requirements and design guidelines
- Signal Integrity simulation options

PCI Express Interfacing

- PCIe Gen3 enhancements
- XILINX PCIe Gen3 Block
- Configuring PCIe interface
- PCIe simulation and implementation
- Debugging PCIe interface
- Electrical requirements and PCB design

Powering

- Power estimation and tools
- Power distribution
- Filtering and decoupling
- Powering realization options

Labs

- Various labs demonstrating IP usage and verifying functionality on UltraScale hardware

XILINX VERSAL™ **PowerWorkshop** Versal ACAP – Connectivity

Die neue Xilinx® Versal™ Adaptive Compute Acceleration Platform (ACAP) erlaubt sehr schnelle Interfaces zu externen Komponenten. Auf den Bausteinen existieren weiterentwickelte Strukturen für die neue Wizards zur Konfiguration in der Entwicklungsumgebung zur Verfügung stehen. Mit der neuen Versal Platform verlagern sich verstärkt Realisierungsherausforderungen vom Chiplevel hin zum Boardlevel. Die sehr hohen Daten-raten erfordern extreme Sorgfalt im PCB-Design. Auch innerhalb der Chips sind neue Strukturen zum Transport der hohen Datenraten implementiert. Der 5-tägige Power-Workshop „Versal ACAP – Connectivity“ richtet sich sowohl an Hardwaredesigner als auch an Systemarchitekten und Layoutdesigner, die schnelle Interfaces erfolgreich im System einsetzen wollen. Mit diesem Workshop wird der Entwickler in die Lage versetzt, die verfügbaren Optionen der IPs und Taktressourcen effektiv einzusetzen. Dabei stehen neben IP Core Konfiguration, Simulation und Implementierung auch die Belange des PCB

Designs und der Signalintegrität im Vordergrund. Anhand eines Versal Boards werden praktische Design- und Verifikationsbeispiele erläutert. Die PCB-Level Simulationsoptionen werden vorgestellt. Im Kurs werden Richtlinien für eine erfolgreiche Realisierung von schnellen Interfaces erarbeitet. Methodische Hinweise zur erfolgreichen Realisierung schneller Interfaces runden den Workshop ab.

Anwendbare Technologien

Versal ACAP

Voraussetzungen

Grundlegende Kenntnisse über VHDL und FPGA-Implementierung

Dauer und Kosten

5 Tage, € 3.100,- netto pro Teilnehmer inklusive ausführlichen Schulungsunterlagen sowie Pausengetränken und Mittagessen

Der Kurs ist auch als 3-Tages Version (€ 2.100) buchbar, hier entfallen die Themen PCI Express und EMAC.

Agenda

Introduction

Architecture Overview and Clocking

- Versal ACAP architecture
- IO architecture and resources
- Clock architecture and resources

Signal Interfacing

- IO standards, termination
- Banking rules
- PCB rules and pin definition

Platform management controller

Network-on-Chip (NoC)

- Introduction and NoC concept
- NoC flow methodology
- NoC performance simulation
- Application examples

Memory Interfacing

- DDR memory controller architecture and functionality
- MC configuration
- Design rules
- Realization of other memory interfaces
- Design practice

Transceivers

- Transceiver overview
- PCS blocks and usage options
- PMA layer options
- Simulation and implementation
- PCB requirements and design guidelines
- Signal Integrity simulation options

PCI Express and CCIX (5-day option only)

- PCIe Gen3/Gen4/Gen5 enhancements
- PL / CPM PCIe blocks
- PCIe simulation and implementation
- DMA and CCIX subsystems
- Debugging and PCB design

Ethernet MAC (5-day option only)

- Multi-rate MAC architecture and functionality
- FEC options
- Application examples

Powering

- Power estimation and tools
- Power distribution, filtering and decoupling
- Powering realization options

Labs

- Various labs deepen the knowledge and allowing practice-oriented training




Ausbildungsschwerpunkt „Business“

Erfolg bedeutet die richtigen Entscheidungen zum richtigen Zeitpunkt zu treffen und die richtigen Fähigkeiten zu besitzen. Ein großes Ziel unserer Schulungen ist das Erwerben von praxisrelevanten, speziellen und theoretischen Kompetenzen, die mit den Anforderungen aus der konkreten Arbeitswelt verzahnt sind.

Mit der Erweiterung unseres Seminarprogrammes um den Bereich „Business“ möchten wir Ihnen ein wichtiges Element zur Seite stellen, um erfolgreich zu sein und zu bleiben, damit Sie sich im richtigen Moment auf Ihre Fähigkeiten verlassen können.

Die in diesem Bereich zusammengefassten Workshops beinhalten Informationen zu den Bereichen „Technik trifft Recht“, „Gesunde Führung und Mindful Leadership“ und „Kommunikation und Feedback“. Denn wie schon der Philosoph und Kommunikationswissenschaftler Paul Watzlawick sagte: „Wir können nicht nicht kommunizieren.“

Die Trainings richten sich vor allem an Führungskräfte, Nachwuchsführungskräfte, Projektleiter und Projektmanager und sollen Ihnen als Werkzeug dienen, um auch die vielfältigen Aufgaben in der Betriebsführung meistern zu können.

Business		Seite	Dauer (Tage)	Kosten
Technik trifft Recht		116	1	€ 750 o. 9 TCs
Gesunde Führung & Mindful Leadership		117	4	€ 2.600 o. 36 TCs
Kommunikation und Feedback		119	2	€ 1.500 o. 18 TCs



Workshop Technik trifft Recht

NEU

Wir leben in einer feinen Welt. Wir sind verwöhnt von Sicherheit, Bequemlichkeit und Unterstützung, die wir durch technische Systeme zur Verfügung gestellt bekommen. Oft sogar ohne unser Wissen. Und dank der Innovationsfreude derer, die sie entwickeln, bekommen wir immer mehr davon. Und grundsätzlich ist unser Vertrauen auf die Zuverlässigkeit technischer Systeme genau richtig.

Zur Realität gehören auch Katastrophen. Dramatisch sind die Auswirkungen auf die Betroffenen und deren Angehörige. Den Verantwortlichen und den schadenverursachenden Unternehmen drohen existenzgefährdende Schadensersatzforderungen.

Zur Katastrophe führen oft mehrere Schadensursachen, die sich zum Teil erst bei der nachträglichen Ursachenforschung vollständig zeigen. Tragisch ist, wenn im Vorfeld bekannte Fehler des technischen Systems nicht kommuniziert werden. Oder die Fehler hätten einfach beseitigt werden können.

Die Zuteilung von Verantwortlichkeiten in einem Schadensfall ist komplex und zeitaufwändig – sowohl technisch, als auch rechtlich.

Im Workshop wird, angelehnt an tatsächliche Unglücksfälle, die rechtliche Haftungssystematik erläutert. Der entstehende lebendige Dialog ermöglicht ein gutes Verständnis für die Thematik und beleuchtet die Wichtigkeit, ein Bewusstsein für das Thema zu haben. Und es werden auch Wege aufgezeigt, die die Wiederholung einer Katastrophe verhindern können.

Zielgruppe:

Führungskräfte, Projektleiter, Projektmanager

Ziele

Einblick in die rechtliche Haftungssystematik
Bewusstsein für Handlungsbedarf entwickeln
Maßnahmen ableiten und umsetzen

Dauer und Kosten

1 Tag, € 750,- netto pro Teilnehmer inklusive ausführlicher Schulungsunterlagen sowie Pausengetränke und Mittagessen

Agenda

- Einführung in die Praxis
- Vorstellung Fallbeispiele aus der Praxis
- Basiswissen zum Aufbau unseres Rechtssystems
- Basiswissen zur rechtlichen Haftungssystematik
- Wer haftet, wenn ein Unglück passiert?
- Wie wird die Haftung unter mehreren Beteiligten aufgeteilt?
- Wer trägt die Beweislast?
- Welche Maßnahmen helfen bei der Fehlervermeidung?

Workshop

NEU

Gesunde Führung und Mindful Leadership

Um ein Unternehmen erfolgreich zu führen braucht es eine Führungskultur mit fähigen Führungskräften. Diese zu entwickeln und erfahrene Vorgesetzte in ihrer Kompetenz als Leader zu fördern ist eine Kernaufgabe. Sie sind der Dreh- und Angelpunkt, um Mitarbeiter zu entwickeln, anzuziehen und an das Unternehmen zu binden. Und sie stehen schlussendlich für die Organisationsentwicklung, tragen Verantwortung für den Unternehmenserfolg.

Sehr vielfältig sind die Aufgaben, vor denen sie stehen. Es braucht eine gute Balance zwischen Einfühlungsvermögen und Durchsetzungskraft, Kompetenz in Kommunikation und Konfliktmanagement, Selbstführung und Fremdführung, sowie unternehmerischem Denken.

Ein „Mindful Leader“ weiß, wie sein persönlicher Führungsstil ist, welche Voraussetzungen für Motivation, Leistung und Erfolg vorliegen müssen. Er initiiert Change-Prozesse und steuert wirksame Mitarbeitergespräche zur Talent-Entwicklung. Er lebt die Ausgewogenheit zwischen Beruf und Freizeit. Er kennt seine Grenzen und seine Werkzeuge zur Selbststeuerung.

Zielgruppe:

Führungskräfte, Nachwuchsführungskräfte

Ziele

Manager oder Führungskraft
Mindful Leadership
Führungsstile
Persönliches Führungsverständnis
Schlüsselaufgaben
Selbstmanagement und Zeitmanagement
Kommunikationskompetenz
Konflikte erkennen, bearbeiten und vermeiden
Mitarbeitergespräche erfolgreich führen
Motivation, Leistung und Erfolg
Balance zwischen beruflichem und privatem Alltag

Dauer und Kosten

4 Tage, € 2.600,- netto pro Teilnehmer inklusive ausführlicher Schulungsunterlagen sowie Pausengetränke und Mittagessen

Agenda

- Basiswissen Führungskraft
- Manager – Führungskraft – Mindful Leadership
- Führungsstile und persönliches Führungsverständnis
- Selbstmanagement
- Zeitmanagement
- Kommunikationskompetenz
- Basiswissen
- Gewaltfreie Kommunikation nach M. Rosenberg
- Empathie im Gespräch
- Mitarbeitergespräche und Feedback
- Konfliktkompetenz
- Konflikte und ihre Bedeutung
- Konfliktbewusstsein – Konfliktwahrnehmung
- Eskalationsstufen und passende Gegenmaßnahmen
- Chef*innen-Sache: Mobbing
- Balance-Akt
- Ausgewogenheit leben
- Selbstwahrnehmung und Selbststeuerung
- Wer oder was bestimmt unser Handeln – gesteuert von der eigenen Software?

Workshop

Kommunikation und Feedback

NEU

„Wir können nicht nicht kommunizieren“ behauptet der Philosoph Paul Watzlawick. Und er hat recht!

Wir sind also ständig und immer in einer Kommunikationssituation, wenn wir anderen begegnen. Und das auch ohne Worte. Sich dessen bewusst zu sein und über die Kommunikation die Möglichkeiten zu nutzen, das berufliche und private Miteinander positiv zu gestalten, erleichtert uns den Alltag. Ist auch Voraussetzung für Erfolg und Entwicklung.

Um es technisch-mechanisch zu formulieren: Innerhalb jeder Kommunikation greifen wie Zahnräder verschiedene Faktoren ineinander. Unsere und die der anderen. Je mehr wir darüber wissen, desto mehr und besser können wir damit umgehen. Und wenn wir wissen, wie eine lösungsorientierte Kommunikation funktioniert, dann können wir auch Konflikte vermeiden oder, wenn sie schon vorhanden sind, diese beenden.

Auch der Empathie-Faktor im Gespräch ist nicht zu unterschätzen. Mit einfachen Mitteln lässt sich hier die Qualität im Miteinander erheblich steigern.

Die positiven Effekte entstehen direkt, sobald die Grundsätze einer guten Kommunikation angewendet werden.

Zur wirksamen Kommunikation gehört auch immer die Fähigkeit, ein wirksames Feedback geben und nehmen zu können. Ganz besonders im beruflichen Alltag hilft eine gut gelebte Feedback-Kultur dabei, die gemeinsame Arbeit stetig zu verbessern und agil in einer sich ständig verändernden Umwelt zu agieren.

Zielgruppe:

alle, insbesondere Führungskräfte

Ziele

Basiswissen zur Kommunikation: „Sie können nicht nicht kommunizieren!“

Kompetenz für lösungsorientierte Kommunikation erwerben

Gewaltfreie Kommunikation für kritische Gespräche

Stolpersteine in der Kommunikation vermeiden

Störfaktoren in Gesprächssituationen entlarven und beseitigen

Eigene „Triggerpunkte“ kennen und damit umgehen

In Projektmeetings und Besprechungen zielorientiert kommunizieren

Erfolgreiche Kommunikation – auch im virtuellen Raum

Feedback – seine Bedeutung und seine Grundlagen

Feedback und seine Regeln für Feedback-Geber*in und Feedback-Nehmer*in

Dauer und Kosten

2 Tage, € 1.500,- netto pro Teilnehmer inklusive ausführlicher Schulungsunterlagen sowie Pausengetränke und Mittagessen

Agenda

- Basiswissen zur Kommunikation
- Sachebene – Beziehungsebene
- Verzerrwinkel in der Kommunikation
- Gewaltfreie Kommunikation nach Marshall Rosenberg
- Was bestimmt unser Handeln – die Sache mit den Triggerpunkten
- Lösungsorientierte Kommunikation in Projekten und Meetings
- Empathie im Gespräch
- Killerphrasen entlarven
- Kommunizieren im virtuellen Raum
- Feedback
- Feedback – seine Bedeutung und seine Grundlagen
- Feedback-Regeln für Feedback-Geber*in und Feedback-Nehmer*in

Seminare

Neben unseren regelmäßig stattfinden Workshops organisiert und veranstaltet PLC2 GmbH sowohl eigenständig als auch in Zusammenarbeit mit ihren Partnern Avnet Silica und EBV Elektronik Seminare zu unterschiedlichen Themeninhalten.

Diese Seminare sind kostenlos und ermöglichen Entwicklern, Teamleitern sowie Mitarbeitern der Führungsebene einen umfassenden Einblick in die vielseitigen Möglichkeiten rund um den Einsatz von programmierbaren Technologien. Unsere erfahrenen Referenten vermitteln kompetent und auf verständliche Art und Weise selbst schwierige Sachverhalte und Zusammenhänge.



Um Ihnen die Wahl eines Seminars zu erleichtern haben wir jedem Seminar ein Symbol mit einem Anforderungsprofil zugeordnet. Das Anforderungsprofil ist in Stufen von 1 bis 9 aufgeteilt und soll Sie bei der Einschätzung von eventuell benötigten Vorkenntnissen unterstützen.

Seminare, die Sie ohne besondere Vorkenntnisse besuchen können, sind mit der Stufe 1 gekennzeichnet.



Weiterführende Seminare, zu denen besondere Vorkenntnisse hilfreich wären, sind mit der Stufe 2 oder höher gekennzeichnet.

Alle Teilnehmer erhalten ausführliche Seminarunterlagen, die auch später noch als Nachschlagewerk verwendet werden können. Selbstverständlich ist der Seminarleiter gerne bereit Ihre Fragen rund um die behandelten Themen zu beantworten.

Lesen Sie auf den nachfolgenden Seiten alle Informationen, die Sie zu den Seminaren benötigen. Sollten weitere Fragen auftauchen, wenden Sie sich an uns, wir werden Ihnen diese gerne beantworten.

Die Seminare werden regelmäßig an verschiedenen Orten durchgeführt. Die aktuellen Orte und Termine entnehmen Sie bitte unserer Homepage unter www.plc2.com.

Seminare	Seite	Dauer (Tage)	Kosten
FPGA Schaltungstechnik	121	1	Kostenlos
PLC2 Basis Seminar Schaltungssynthese mit VHDL	122	1	Kostenlos
PLC2 Basis Seminar Schaltungssimulation mit VHDL	123	1	Kostenlos



PLC2 Basis Seminar FPGA Schaltungstechnik

Das Seminar zum digitalen Schaltungsentwurf – insbesondere für die FPGA Technologie – ist der erste Teil einer Reihe von kostenlosen PLC2 Basis Seminaren.

In der allgemeinen Einführung zu „FPGA Schaltungstechnik“ werden die grundlegenden Schaltungselemente eines FPGA besprochen. Dies umfasst kombinatorische und sequentielle Grundsaltungen wie Multiplexer, Lookup-Tables, Flip-Flops, RAMs, Addierer, Multiplizierer, Taktgeneratoren und I/Os. Anschließend werden komplexere Grundsaltungen wie Vergleicher, Zähler, Schieberegister, FIFO, Stack und Zustandsautomaten aus Basiselementen aufgebaut.

Um weitere theoretische und praktische Grundlagen zu schaffen, werden Designs mit mehreren Taktnetzwerken und die Datenübergabe bzw. Synchronisierung zwischen verschiedenen Taktdomänen (Clock-Domain-Crossing – CDC) betrachtet. Ergänzend werden verschiedene

Synchronisierungsschaltungen vorgestellt. Dies schließt fernerhin eine Betrachtung von Propagation-Delays sowie Setup- und Hold-Zeiten mit ein.

Im abschließenden Kapitel des Seminars werden ausgewählte Datenaustauschprotokolle zur Kopplung von Grundsaltungen, IP Cores oder auch Prozessoren besprochen.

Anwendbare Technologien

Alle Xilinx FPGAs inklusive SoCs

Voraussetzungen

Keine

Dauer und Kosten

1 Tag, die Teilnahme an diesem Seminar ist kostenlos, Seminarsprache ist Deutsch

Agenda

Generische FPGA Architektur

- Primitive Schaltungselemente des FPGA
- Interne Verdrahtung, Verschaltungsmatrizen
- Dedizierte Arithmetikblöcke und Speicher
- I/O Ressourcen und Taktnetzwerke
- Integrierte IP Blöcke
- SoC (Prozessoren, FPGA, Peripherie)

Grundsaltungen

- Flip-Flop, Register, Schieberegister
- Addierer, Zähler
- RAM, ROM
- FIFO, Stack
- Zustandsautomat (FSM)

Clock-Domain-Crossing

- Propagation Delay
- Setup-/Hold-Time
- 2-FF Synchronizer
- Strobe Synchronizer
- Vector Synchronizer

Datenaustauschprotokolle

- Handshake Signale
- Streaming Protokolle



PLC2 Basis Seminar Schaltungssynthese mit VHDL

Das Seminar zur Schaltungssynthese mit VHDL ist Teil einer Reihe von kostenlosen PLC2 Basis Seminaren. Programmierbare Logikbausteine, wie FPGAs, haben sich in allen Bereichen unseres täglichen Lebens etabliert. Sie werden z.B. in mobilen Telefonen, IoT-Geräten, Automobilen oder Rechenzentren eingesetzt. Ihre Einsatzbereiche sind dabei so vielfältig wie ihre Größe. Sie dienen dabei als Protokolladapter, als Signalkonverter oder als Beschleuniger für die Video-, Radar- und Sensordatenauswertung.

Der Entwurf digitaler Schaltungen dieser Größenordnung bedingt den Einsatz einer leistungsfähigen Hardwarebeschreibungssprache, die durch diverse Abstraktionsmechanismen den Entwickler in die Lage versetzt, schnell und effektiv ein Hardwaredesign zu erstellen. VHDL erfüllt all diese Anforderungen. VHDL ist eine stark typisierte Hardwarebeschreibungssprache, die bereits während des Entwurfs verschiedenfältige Programmierfehler ausschließt. Typischerweise wird VHDL auf dem Register-Transfer-Level (RTL) eingesetzt, um digitale Schaltungen beliebiger Komplexität zu entwerfen. Darüber hinaus kann VHDL zur Integration größerer Teil-

schaltungen auf Systemebene eingesetzt werden. Neben den VHDL Sprachkonstrukten für die Synthese bietet die Sprache diverse Funktionalitäten, um komplexe Verifikationsmodelle zu beschreiben. Damit ist es möglich, digitale Schaltungen vom simplen Gatter bis zum System-on-Chip (SoC) vorab zu verifizieren. In diesem Seminar werden ausgewählte Aspekte der Hardwarebeschreibungssprache VHDL, basierend auf der Sprachrevision IEEE Std. 1076-2008, behandelt.

Anwendbare Technologien

Alle Xilinx FPGAs inklusive SoCs

Voraussetzungen

Ein grundlegendes Verständnis der digitalen Schaltungstechnik, sowie Konzepte einer beliebigen Programmier- oder Skriptingsprache sind wünschenswert.

Dauer und Kosten

1 Tag, die Teilnahme an diesem Seminar ist kostenlos, Seminarsprache ist Deutsch

Agenda

Einführung in VHDL

- Historie und Zukunft von VHDL
- Sprachkonzept
- Hardware Modellierungstechniken
- Designflow

Hardwarebeschreibung mit VHDL

- Entity / Architecture
- Package
- Library / Context

Sprachelemente

- Signale, Variablen, Konstanten
- Prozesse und Nebenläufigkeit
- Kontrollstrukturen
- Funktionen / Prozeduren
- Generische Beschreibungen

Starke Typisierung in VHDL

- Vordefinierte Typen und Operatoren
- Nutzerdefinierte Typen
- Attribute



PLC2 Basis Seminar

Schaltungssimulation mit VHDL

Das Seminar zur Schaltungssimulation mit VHDL ist Teil einer Reihe von kostenlosen PLC2 Basis Seminaren .

Programmierbare Logikbausteine, wie FPGAs, haben sich in allen Bereichen unseren täglichen Lebens etabliert. Sie werden z.B. in mobilen Telefonen, IoT-Geräten, Automobilen oder Rechenzentren eingesetzt. Ihre Einsatzbereiche sind dabei so vielfältig wie ihre Größe. Sie dienen dabei als Protokolladapter, als Signalkonverter oder als Beschleuniger für die Video-, Radar und Sensordatenauswertung.

Der Entwurf digitaler Schaltungen dieser Größenordnung bedingt den Einsatz einer leistungsfähigen Hardwarebeschreibungssprache, die durch diverse Abstraktionsmechanismen den Entwickler in die Lage versetzt, schnell und effektiv ein Hardwaredesign zu erstellen. VHDL erfüllt all diese Anforderungen.

VHDL ist eine stark typisierte Hardwarebeschreibungssprache, die bereits während des Entwurfs verschiedenfältige Programmierfehler ausschließt. Typischerweise wird VHDL auf dem Register-Transfer-Level (RTL) eingesetzt, um digitale Schaltungen beliebiger Komplexität zu entwerfen. Darüber hinaus kann VHDL zur Integration größerer Teilschaltungen auf Systemebene eingesetzt

werden. Neben den VHDL Sprachkonstrukten für die Synthese bietet die Sprache diverse Funktionalitäten, um komplexe Verifikationsmodelle zu beschreiben. Damit ist es möglich, digitale Schaltungen vom simplen Gatter bis zum System-on-Chip (SoC) vorab zu verifizieren.

In diesem Seminar werden unter Rückgriff auf bereits erlerntes VHDL Wissen, die erweiterten Sprachkonstrukte zum Erstellen von Simulationen nach dem VHDL Testbench Konzept behandelt. Die vermittelten Sprachelemente der Hardwarebeschreibungssprache VHDL basieren auf der Sprachrevision IEEE Std. 1076-2008.

Anwendbare Technologien

Alle Xilinx FPGAs inklusive SoCs

Voraussetzungen

Grundlagen in der Hardwarebeschreibungssprache VHDL, z.B. aus dem PLC2 Basis Seminar Schaltungssynthese mit VHDL sind wünschenswert.

Dauer und Kosten

1 Tag, die Teilnahme an diesem Seminar ist kostenlos, Seminarsprache ist Deutsch

Agenda

VHDL Rückblick

- Prozesse und Nebenläufigkeit
- Wait Statements, Delta-Cycles

Testbench Konzept

- Inline Testbenches
- Modulare Testbenches
- Assertions
- Stimuli und Checks

Einfache Stimuli Erzeugung

- Einfache Stimuli Erzeugung
- Analoge Stimuli
- Zufällige Stimuli
- Self-Checking Testbench

File I/O

- Lesen und Schreiben von Dateien
- Logging

Stand Oktober 2021 – Änderungen vorbehalten

Embedded	Seite	Dauer (Tage)	Januar	Februar	März	April
ZYNQ UltraScale + MPSoC System Architecture	54	2			30.-31. Stuttgart	
Comp. ZYNQ UltraScale+ MPSoC HW Designers	55	3		02.-04. Frankfurt		
Adv. ZYNQ UltraScale+ MPSoC HW Designers	56	3			21.-23. München	
Comp. ZYNQ UltraScale+ MPSoC SW Designers	57	3		14.-16. Stuttgart		
Professional ZYNQ UltraScale + MPSoC	58	5			07.-11. Freiburg	
ZYNQ-7000 SoC System Architecture	59	2		16.-17. Frankfurt		
Compact ZYNQ-7000 SoC for HW Designers	60	3			14.-16. Stuttgart	
Compact ZYNQ-7000 SoC for SW Designers	61	3		07.-09. München		
Professional ZYNQ-7000 SoC	62	5	24.-28. Freiburg			04.-08. Freiburg
Expert ZYNQ-7000 SoC	63	5	Auf Anfrage			
Compact VITIS for Software Designers	64	3		09.-11. Freiburg		
Compact VITIS for Acceleration	65	3		21.-23. München		
VITIS HLS	66	3		21.-23. Frankfurt		
Compact VITIS AI	67	3			28.-30. Frankfurt	
Professional VITIS	68	5			21.-25. Frankfurt	
Versal ACAP System Architecture	69	2	17.-18. Freiburg			11.-12. Frankfurt
Compact Versal ACAP for HW Designer	70	3	19.-21. Freiburg			20.-22. Frankfurt
Compact Versal ACAP for SW Designer	71	3		01.-03. Stuttgart		
Professional Versal ACAP	72	5			07.-11. München	
Advanced Versal AI Engine	73	3			02.-04. Stuttgart	
Expert Versal AI Engine	74	5	24.-28. Stuttgart			25.-29. Freiburg
AXI Interface Technology	75	2			03.-04. Freiburg	
High-Speed Ethernet – Hands-On System Dev.	76	3	Auf Anfrage			
PCI Express – Hands-On System Development	77	3			02.-04. Freiburg	
Designing with RF Data Converters	78	3		01.-03. Berlin		
Professional RFSoc	79	5	Auf Anfrage			
Professional MicroBlaze System Design	82	5	Auf Anfrage			
Dev. Multimedia Solutions VCU and GStreamer	83	2		17.-18. Stuttgart		13.-14. Freiburg
Compact Embedded Linux	84	3	Auf Anfrage			
Embedded Linux Development Yocto Project	85	2	Auf Anfrage			
Embedded Linux Treiberentwicklung	86	3	Auf Anfrage			
Embedded Design with PetaLinux Tools	87	2			24.-25. Stuttgart	
Compact Python for Embedded	88	3			07.-09. Stuttgart	
Professional Python for Embedded	89	5			07.-11. Stuttgart	

Mai	Juni	Juli	August	September	Oktober	November	Dezember	Kosten
	13.-14. Frankfurt			15.-16. Freiburg			01.-02. München	€ 1.500 o. 18 TCs
23.-25. Freiburg			23.-25. Berlin			21.-23. München		€ 2.100 o. 27 TCs
	08.-10. Berlin			28.-30. Frankfurt			19.-21. Stuttgart	€ 2.100 o. 27 TCs
18.-20. Frankfurt			09.-11. Berlin			14.-16. Freiburg		€ 2.100 o. 27 TCs
	20.-24. Frankfurt			19.-23. Freiburg			12.-16. Frankfurt	€ 3.100 o. 45 TCs
	07.-08. Berlin			08.-09. Freiburg		21.-22. München		€ 1.500 o. 18 TCs
	13.-15. Freiburg			19.-21. Frankfurt			12.-14. München	€ 2.100 o. 27 TCs
	27.-29. Berlin			12.-14. Freiburg		02.-04. Frankfurt		€ 2.100 o. 27 TCs
		11.-15. Freiburg			24.-28. Freiburg			€ 3.100 o. 45 TCs
			Auf Anfrage					€ 3.100 o. 45 TCs
11.-13. Frankfurt				20.-22. Berlin			05.-07. München	€ 2.100 o. 27 TCs
	17.-19. Berlin			01.-03. Freiburg		14.-16. Frankfurt		€ 2.100 o. 27 TCs
18.-20. München				29.-31. Freiburg		07.-09. Stuttgart		€ 2.100 o. 27 TCs
	01.-03. Freiburg				26.-28. Stuttgart	28.-30. Berlin		€ 2.100 o. 27 TCs
	20.-24. Stuttgart			08.-12. Freiburg		07.-11. Berlin		€ 3.100 o. 45 TCs
				01.-02. Stuttgart		23.-24. München		€ 1.500 o. 18 TCs
		27.-29. München			12.-14. Stuttgart			€ 2.100 o. 27 TCs
23.-25. Frankfurt				03.-05. Freiburg		05.-07. Berlin		€ 2.100 o. 27 TCs
	20.-24. Stuttgart				12.-16. Frankfurt		07.-11. Freiburg	€ 3.100 o. 45 TCs
	08.-10. Frankfurt				05.-07. Freiburg			€ 2.100 o. 27 TCs
		18.-22. München				17.-21. Frankfurt		€ 3.100 o. 45 TCs
	29.-30. Frankfurt				12.-13. Berlin		29.-30. München	€ 1.500 o. 18 TCs
			Auf Anfrage					€ 2.100 o. 27 TCs
	01.-03. Berlin				12.-14. Frankfurt		05.-07. Stuttgart	€ 2.100 o. 27 TCs
17.-19. Freiburg					19.-21. Stuttgart		12.-14. München	€ 2.100 o. 27 TCs
			Auf Anfrage					€ 3.100 o. 45 TCs
			Auf Anfrage					€ 3.100 o. 45 TCs
		13.-14. Frankfurt				06.-07. Berlin		€ 1.500 o. 18 TCs
			Auf Anfrage					€ 2.100 o. 27 TCs
			Auf Anfrage					€ 1.500 o. 18 TCs
			Auf Anfrage					€ 2.100 o. 27 TCs
	29.-30. Frankfurt				14.-15. Berlin		29.-30. München	€ 1.500 o. 18 TCs
	20.-22. Freiburg				05.-07. Frankfurt		21.-23. Berlin	€ 2.100 o. 27 TCs
	20.-24. Freiburg				05.-09. Frankfurt		21.-25. Berlin	€ 3.100 o. 45 TCs




Stand Oktober 2021 – Änderungen vorbehalten

Stand Oktober 2021 – Änderungen vorbehalten

Embedded	Seite	Dauer (Tage)	Januar	Februar	März	April
Migrate to Real-Time OS: Hands-On Syst. Dev.	90	2			Auf Anfrage	
Real-Time Control Sys. Dev. using RTOS	91	2			Auf Anfrage	
Essentials of Microprocessors	93	1			Auf Anfrage	
RISC-V Architecture and FPGA implementation	94	2			Auf Anfrage	
RISC-V Core Verification & Compliance Testing	95	2			Auf Anfrage	

DSP & Image Processing	Seite	Dauer (Tage)	Januar	Februar	März	April
DSP Design using System Generator	98	3		07.-09. München		25.-27. Frankfurt
Model Composer	99	2		10.-11. München		28.-29. Frankfurt
Expert DSP Design	100	5		07.-11. München		25.-29. Frankfurt

Connectivity	Seite	Dauer (Tage)	Januar	Februar	März	April
High-Speed Memory Interfacing	103	3			14.-16. Berlin	
DDR4 Interfacing with Xilinx FPGAs	104	3		14.-16. Berlin		
Designing with Multi-Gigabit Serial I/O	105	3	25.-27. München			
Designing with Ethernet MAC Controllers	106	2	19.-20. München			
Designing with PCI Express	107	3		07.-09. Stuttgart		
Professional PCI Express	108	5				04.-08. Berlin
Signal Integrity	109	3	17.-19. Frankfurt			
FPGA Board Design	110	2	20.-21. Frankfurt			
ZYNQ Board Design & High-Speed Interfacing	111	3		21.-23. Frankfurt		
UltraScale Connectivity	112	5			07.-11. München	
VERSAL ACAP Connectivity	113	5			21.-25. Frankfurt	

Business	Seite	Dauer (Tage)	Januar	Februar	März	April
Technik trifft Recht 	116	1			Auf Anfrage	
Gesunde Führung & Mindful Leadership 	117	4			Auf Anfrage	
Kommunikation und Feedback 	119	2			Auf Anfrage	

Seminare	Seite	Dauer (Tage)	Januar	Februar	März	April
FPGA Schaltungstechnik	121	1			29. München	
Schaltungssynthese mit VHDL	122	1				26. Frankfurt
Schaltungssimulation mit VHDL	123	1				

Mai	Juni	Juli	August	September	Oktober	November	Dezember	Kosten
								€ 1.500 o. 18 TCs
								€ 1.500 o. 18 TCs
								€ 750 o. 9 TCs
								€ 1.500 o. 18 TCs
								€ 1.500 o. 18 TCs

Mai	Juni	Juli	August	September	Oktober	November	Dezember	Kosten
				05.-07. Freiburg			05.-07. Berlin	€ 2.100 o. 27 TCs
				08.-09. Freiburg			08.-09. Berlin	€ 1.500 o. 18 TCs
				05.-09. Freiburg			05.-09. Berlin	€ 3.100 o. 45 TCs

Mai	Juni	Juli	August	September	Oktober	November	Dezember	Kosten
	13.-15. München					02.-04. Frankfurt		€ 2.100 o. 27 TCs
		18.-20. Frankfurt					19.-21. Stuttgart	€ 2.100 o. 27 TCs
04.-06. Berlin				05.-07. Frankfurt				€ 2.100 o. 27 TCs
23.-24. Stuttgart				14.-15. Berlin		17.-18. Freiburg		€ 1.500 o. 18 TCs
23.-25. Freiburg				12.-14. München			05.-07. Berlin	€ 2.100 o. 27 TCs
			01.-05. Frankfurt			14.-18. Freiburg		€ 3.100 o. 45 TCs
	20.-22. Berlin					21.-23. Stuttgart		€ 2.100 o. 27 TCs
	23.-24. Berlin					24.-25. Stuttgart		€ 1.500 o. 18 TCs
10.-12. München				26.-28. Berlin				€ 2.100 o. 27 TCs
		25.-29. Frankfurt			24.-28. Berlin			€ 3.100 o. 45 TCs
		11.-15. Berlin				07.-11. Stuttgart		€ 3.100 o. 45 TCs

Mai	Juni	Juli	August	September	Oktober	November	Dezember	Kosten
								€ 750 o. 9 TCs
								€ 2.600 o. 36 TCs
								€ 1.500 o. 18 TCs

Mai	Juni	Juli	August	September	Oktober	November	Dezember	Kosten
				20. Freiburg				kostenlos
					18. Stuttgart			kostenlos
10. Freiburg						15. Frankfurt		kostenlos

Stand Oktober 2021 – Änderungen vorbehalten

Allgemeine Geschäftsbedingungen der PLC2 GmbH (nachfolgend: PLC2)

§ 1 Geltungsbereich

- Die Dienstleistungs- und Lieferverträge von PLC2 werden ausschließlich zu den nachstehenden Allgemeinen Geschäftsbedingungen abgeschlossen.
- Etwaige Einkaufs-, Beschaffungs- und sonstige Allgemeine Geschäftsbedingungen des Vertragspartners finden selbst dann keine Anwendung, wenn der Vertragspartner in seiner Bestellung hierauf Bezug nimmt und PLC2 nicht nochmals ausdrücklich widerspricht.

§ 2 Anmeldung und Vertragsschluss

- Anmeldungen für Schulungen können formlos, schriftlich, per Fax oder über ein Online-Formular vorgenommen werden.
- Angebote von PLC2 sind, soweit nicht einzelvertraglich anders geregelt, freibleibend. Ein Vertrag kommt erst durch die Unterschrift beider Vertragspartner oder durch einen schriftlichen Auftrag des Vertragspartners und seiner Annahme seitens PLC2 durch Gegenzeichnung oder schriftliche Bestätigung zustande. Ausschließlich der so bestätigte Vertragsinhalt und ergänzend diese AGB sind Grundlage für die Leistungserbringung durch PLC2. Der Umfang der vertraglichen Leistungen ergibt sich aus den vorgenannten Vertragsdokumenten und den zugehörigen Anlagen.

§ 3 Art und Umfang der Leistung

- PLC2 bietet offene Schulungen, sowie Firmentrainings in Räumen des Vertragspartners und in den mit dem Vertragspartner vereinbarten Räumen an.
- Der genaue Veranstaltungsort, sowie der Termin, werden jeweils in der Auftragsbestätigung angegeben.
- Bei Firmentrainings werden die Leistungsdetails einzelvertraglich festgelegt. Bei Schulungen im Hause des Vertragspartners stellt dieser die notwendige und geeignete Infrastruktur, insbesondere Seminarräumlichkeiten und Seminarausstattung, installierte Software und Hardware, Zugriffsrechte zu Verfügung. PLC2 stellt auf Nachfrage eine Liste über die notwendigen Voraussetzungen dem Vertragspartner rechtzeitig zur Verfügung.
- Der Vertragspartner kann PLC2 mit nachträglichen Änderungen in Inhalt und Umfang vereinbar Leistungen beauftragen, sofern dies für PLC2 zumutbar ist und, falls andere vertragliche Regelungen (z.B. Preise, Termine) von derartigen Änderungen berührt werden, auch hierüber eine Einigung erzielt worden ist. Bis zum Abschluss einer solchen Vereinbarung wird PLC2 die Schulung nach der bisherigen Vereinbarung fortsetzen.
- PLC2 behält sich das Recht vor, Leistungen zu erweitern, zu ändern und zu verringern, soweit der Vertragsgegenstand dadurch nicht wesentlich verändert wird und dies für den Vertragspartner zumutbar ist. Das Recht zur Leistungsänderung steht PLC2 insbesondere dann zu, wenn diese Änderung handelsüblich oder PLC2 hierzu, durch Änderung der Gesetzeslage oder durch die Rechtsprechung, verpflichtet ist.
- Jeder Teilnehmer erhält für seine Schulungsteilnahme ein auf ihn persönlich ausgestelltes Teilnahmezertifikat.
- Für die Reise zum Veranstaltungsort und ggf. Buchung von Hotelzimmern o.ä. ist grundsätzlich der Vertragspartner selbst verantwortlich. Wird eine Hotelreservierung für Übernachtungen gewünscht, so ist dies bei Anmeldung – mindestens jedoch zwei Wochen vor Kursbeginn – mitzuteilen. Gewünschte Hotelreservierungen werden von PLC2 im Namen und auf Rechnung des Vertragspartners vorgenommen.

§ 4 Stornierung und Umbuchung

- Die Stornierung einer Anmeldung muss schriftlich erfolgen.
- Stornierungen und Umbuchungen eines Vertragspartners für eine offene Schulung sind bis zwei Wochen vor Veranstaltungsbeginn kostenlos.
- Stornierungen und Umbuchungen eines Vertragspartners für ein Firmentraining sind bis vier Wochen vor Veranstaltungsbeginn kostenlos.
- Bei einer Stornierung nach den oben genannten Zeitpunkten oder bei Nichtteilnahme werden 100% des Schulungspreises berechnet.
- Bei Umbuchung auf eine gleichwertige Schulung zu einem späteren Zeitpunkt, in Absprache mit PLC2, entstehen keine Kosten.
- Anstelle einer Stornierung oder Umbuchung hat der Vertragspartner das Recht, ohne Mehrkosten einen Ersatzteilnehmer zu der von ihm gebuchten Schulung zu schicken.
- PLC2 behält sich vor, eine Schulung bis ca. sieben Tage vor Beginn der Schulung aus wichtigen Gründen, z.B. bei zu geringer Teilnehmerzahl oder bei Ausfall eines Referenten (dann u.U. auch kurzfristig), höherer Gewalt oder gleichzeitiger Gründe, abzusagen. In jedem Fall ist PLC2 bemüht, Absagen oder notwendige Änderungen des Programms, insbesondere eines Dozentenwechsels, so rechtzeitig wie möglich mitzuteilen.
- Muss eine Schulung abgesagt werden, so bietet PLC2 dem Vertragspartner die Möglichkeit, am nächstmöglichen Schulungstermin einer gleichartigen Schulung teilzunehmen. Sollte der Vertragspartner an diesem Termin nicht teilnehmen, so kann er sich eine andere Schulung im gleichen Preisniveau auswählen und umbuchen.
- Bei Absage einer Schulung durch PLC2 sind weitere Ansprüche gegen PLC2 ausgeschlossen, außer in Fällen vorsätzlichen oder grob fahrlässigen Verhaltens von Angestellten der PLC2 oder deren Erfüllungsgehilfen.

§ 5 Rechte an den Schulungsunterlagen, Software und Übungen

PLC2 behält sich alle Rechte an den Schulungsunterlagen, der von ihr gestellten Software und den Übungseinheiten vor. Schulungsunterlagen, gestellte Software und Übungseinheiten dürfen ohne schriftliche Zustimmung, auch auszugsweise oder zu Unterrichtszwecken, weder reproduziert noch verändert, übersetzt – insbesondere unter Verwendung elektronischer Systeme – verarbeitet, vervielfältigt, verbreitet oder zur öffentlichen Weitergabe benutzt werden. Der Vertragspartner von PLC2 wird alle Benutzer von autorisierten Kopien – Programme und Dokumentation – über die Autorisierung zur Verteilung und Nutzung von Produkten von PLC2 in Kenntnis setzen und sie verpflichten, diese Bedingungen ebenfalls einzuhalten. Der Vertragspartner verpflichtet sich, Aufzeichnungen zu führen, die nachweisen, dass er die Bedingungen eingehalten hat. Auf Anfrage wird er PLC2 oder einer von PLC2 beauftragten unabhängigen Organisation diese Aufzeichnungen zu Prüfzwecken zur Verfügung stellen.

§ 6 Haftungsumfang und Nutzung IT-Infrastruktur

- Die Schulungen von PLC2 werden von deren Dozenten mit Sachkunde und größtmöglicher Sorgfalt vorbereitet und geleitet. PLC2 haftet, gleichwohl aus welchem Rechtsgrund, für durch Angestellte der PLC2 oder deren Erfüllungsgehilfen vorsätzlich oder grob fahrlässig verursachte Schäden einmalig bis zur Höhe des entrichteten Schulungspreises. Eine weitergehende Haftung, sowie jede Haftung für mittelbare Folgeschäden ist ausgeschlossen.
- Für den Verlust oder die Beschädigung des Eigentums des Bestellers oder von Dritten sowie für Unfälle kann PLC2 keine Haftung übernehmen.
- Es wird ausdrücklich vereinbart, dass ein eventuell bereitgestellter Internetzugang nicht für Nutzungen außerhalb des Schulungsrahmens verwendet werden darf. Insbesondere verpflichtet sich jeder Teilnehmer folgendes zu unterlassen:
 - Ins-Netz-Stellen oder Abrufen von Dateien, die gegen Datenschutzrechtliche, persönlichkeitsrechtliche, unethische oder strafrechtliche Bestimmungen verstoßen,
 - von beleidigenden, verleumdenden, verfassungsverfehlenden, rassistischen, sexistischen Äußerungen oder Abbildungen.
 - das Ausprobieren, das Ausforschen und die unberechtigte Benutzung fremder Zugriffsberechtigungen (wie z. B. Benutzererkennungen, Passworte) und sonstiger Authentifizierungsmittel (wie z. B. Chipkarten, Magnetkarten) ist unzulässig,
 - die Weitergabe und das Zurverfügungstellen von eigenen Benutzererkennungen und sonstigen Authentifizierungshilfsmitteln für eine Benutzung durch Dritte ist unzulässig. Es wird ausdrücklich darauf hingewiesen, dass in einem derartigen Fall aus den Protokolldaten die Identität des jeweiligen Teilnehmers hergeleitet. Jegliche Aktivität – auch unzulässige – durch diesen Dritten wird also dem jeweiligen Teilnehmer zugeschrieben.
- Von Teilnehmern mitbrachte Datenatäger dürfen nicht auf den Rechnern von PLC2 benutzt werden. Bei Zuwiderhandlung behält sich PLC2 Schadensersatzforderungen vor.
- In den Schulungen und Trainings wird Software eingesetzt, die urheberrechtlich geschützt ist. Diese Software und deren Dokumentation darf weder aus den Veranstaltungsräumen entfernt, noch ganz oder teilweise kopiert oder auf andere, nicht genehmigte Weise nutzbar gemacht werden.
- Nutzungsrechte an Softwareprodukten von PLC2 oder Dritten sind in dem für den Schulungsbetrieb notwendigen Umfang für die Dauer der Schulung im Schulungspreis enthalten.

§ 7 Vergütung und Zahlungsverzug

- Es gelten die im aktuellen Workshop- und Seminarprogramm von PLC2 angegebenen Preise. Das jeweils aktuelle Programmheft ist bei PLC2 anzufordern. Alle dort angegebenen Preise verstehen sich zuzüglich der aktuell gültigen, gesetzlichen Mehrwertsteuer.
- Die Workshop- und Seminarpreise für offene Schulungen sind inklusive Pausengetränke, Mittagessen und ausführlichen Trainingsunterlagen.
- Rechnungsbeträge sind ungekürzt und unter Angabe der Rechnungsnummer 14 Tage nach Rechnungsstellung, jedoch vor Schulungsbeginn, zu begleichen.
- Sofern der Vertragspartner in Verzug gerät werden eingehende Zahlungen zunächst auf eventuelle Kosten und Zinsen, sodann auf die älteste Forderung verrechnet.
- Eine nicht durch PLC2 zu vertretende nur zeitweise Teilnahme an Schulungen berechtigt nicht zur Vergütungsminderung.

§ 8 Datenschutz

PLC2 wird Daten im Sinne des Bundesdatenschutzgesetzes, der aktuell gültigen Corona-Verordnung und anderer einschlägiger Datenschutzbestimmungen nur nach Maßgabe der jeweils einschlägigen gesetzlichen Bestimmungen erheben, verarbeiten und nutzen.

§ 9 Allgemeine Bestimmungen

- Rechtswahl:** Hiernach geschlossene Verträge unterliegen deutschem Recht unter Ausschluss einer etwaigen Weiterverweisung auf ausländisches Recht. Das UN-Übereinkommen über Verträge über den internationalen Warenkauf (United Nations Convention on Contracts for the International Sale of Goods, CISG) findet keine Anwendung.
- Erfüllungsort, Gerichtsstand:** Erfüllungsort ist der Sitz der PLC2. Gerichtsstand ist soweit gesetzlich zulässig der Sitz der PLC2.
- Rechte- und Pflichtenübertragung:** Die Parteien sind unter vorheriger schriftlichen Anzeige berechtigt, die Rechte und Pflichten aus dem Vertrag auf mit ihnen verbundene Unternehmen ganz oder teilweise zu übertragen. Im Übrigen bedarf eine Übertragung der Rechte und Pflichten aus dem Vertrag der schriftlichen Einwilligung der jeweils anderen Partei. Diese wird ihre Einwilligung nicht unbillig verweigern. Die vorstehenden Regelungen gelten nicht für die Abtretung von Geldforderungen.
- Schriftform:** Änderungen und Ergänzungen eines Vertrages bedürfen der Schriftform. Ein Verzicht hierauf kann nur schriftlich erfolgen. Das Schriftformerfordernis kann auch durch Telefax oder durch Briefwechsel gewahrt werden. § 127 Abs. 2 S. 1 BGB findet im Übrigen jedoch keine Anwendung.
- Vertrags Erfüllung durch Dritte:** Die PLC2 ist berechtigt, zur Erfüllung der ihr nach hiernach abgeschlossener Verträge obliegende Verpflichtungen Dritte, einschließlich mit ihr im Sinne von §§ 151 ff. AktG verbundene Unternehmen, einzuschalten.
- Konzerndelegationsrecht:** Die PLC2 ist berechtigt alle Rechte und Pflichten aus dem Vertrag an ein mit ihr i.S.d. §§ 15 ff. AktG verbundenes Unternehmen abzutreten.
- Salvatorische Klausel:** Erweist sich eine Bestimmung dieser Allgemeinen Geschäftsbedingungen oder eines hiernach geschlossenen Vertrages ganz oder teilweise als unwirksam oder nicht durchsetzbar, so berührt dies die Wirksamkeit und Durchsetzbarkeit der übrigen Bestimmungen dieser Allgemeinen Geschäftsbedingungen oder des betreffenden Vertrages nicht.
- Höhere Gewalt:** Mit Ausnahme der Verpflichtung zur Vornahme von Zahlungen ist jede Partei von der Erfüllung ihrer hiernach abgeschlossenen Verträge bestehenden Vertragspflichten solange befreit, als diese infolge höherer Gewalt unmöglich ist. Höhere Gewalt sind insbesondere Streik oder rechtmäßige Absperrung, Feuer, Überflutung, Epidemien, Pandemien, Quarantäneanordnungen, behördliche Maßnahmen, Verzug oder Nichterfüllung seitens Zulieferanten, Erdbeben, Ausfall von und Störungen in Kommunikationsnetzen oder andere von der jeweils leistungswilligen Partei nicht zu vertretende Umstände.