

Effective Coding With Vhdl Principles And Best Practice The Mit Press

Die Anforderungen der Benutzer an die Systeme sind in den letzten Jahren kontinuierlich gestiegen. Hinzu kommt, dass immer mehr Software-Produkte - insbesondere auch das Betriebssystem Windows selbst - die grafische Darstellung nicht mehr auf der CPU (Central Processing Unit) berechnen, sondern auf der hierfür optimierten GPU (Graphics Processing Unit). Steht keine GPU zur Verfügung, erfolgt die Berechnung weiterhin auf der CPU. Da die grafische Darstellung aber immer aufwendiger wird, ist die CPU-Belastung hierdurch sehr deutlich spürbar. Gerade beim Einsatz von virtuellen Desktops - also Desktops die im Rechenzentrum bereitgestellt werden - ist die fehlende GPU oft ein deutlicher Nachteil gegenüber klassischen PCs. Zudem steigt die Belastung der entsprechenden Server deutlich und die Anzahl der Benutzer pro physikalischem Server sinkt. Die Kosten pro Benutzer steigen also. Die Industrie hat diesen Nachteil bereits vor einigen Jahren erkannt und die GPU-Passthrough Technologie eingeführt. Hierbei wird eine physikalische Grafikkarte einer virtuellen Maschine zugeordnet. Diese kann dann die volle Leistungsfähigkeit der Grafikkarte nutzen. Jede virtuelle Maschine benötigt hierbei allerdings ihre eigene Grafikkarte. NVIDIA hat die GRID Technologie veröffentlicht. Hiermit wird es möglich, dass mehrere virtuelle Maschinen sich eine Grafikkarte teilen, aber dennoch die volle Leistung erhalten können - ähnlich wie dies bereits in allen virtuellen Umgebungen mit der CPU passiert. Sie erfahren, was Sie bei der Planung einer GPU unterstützten Umgebung beachten müssen und lernen Schritt für Schritt, wie Sie diese mit Citrix und NVIDIA bereitstellen. Anschließend folgen die Überwachung der Umgebung und die Behebung bekannter Fehler. Inhalte (Auszug): Plan (v)GPU Typen Lizenz-Varianten HDX / HDX 3D Pro Do Grafikkarten Lizenz Server Virtual Delivery Agent Check Monitoring GPU Check GPU Monitoring NVIDIA-SMI GPU Profiler Act Systemlast 3Dconnexion Geräte Troubleshooting

Nach ihrem Sieg im berühmten Turnier von Badminton stehen für Casey die Zeichen auf Erfolg. Was läge näher, als gleich für die nächste Stufe des Grand Slam, das Kentucky Derby, zu trainieren? Aber jetzt gerät Caseys Vater in Verdacht. Und ihr Freund Peter scheint Zweifel an ihm zu haben. Zu allem Überfluss ist auch die Harmonie zwischen Casey und ihrem geliebten Pferd Storm Warning empfindlich gestört. Aber Casey gibt so schnell nicht auf. Ob Naturwissenschaftler, Mathematiker, Ingenieur oder Datenwissenschaftler - mit MATLAB haben Sie ein mächtiges Tool in der Hand, das Ihnen die Arbeit mit Ihren Daten erleichtert. Aber wie das mit manch mächtigen Dingen so ist - es ist auch ganz schön kompliziert. Aber keine Sorge! Jim Sizemore führt Sie in diesem Buch Schritt für Schritt an das Programm heran - von der Installation und den ersten Skripten bis hin zu aufwändigen Berechnungen, der Erstellung von Grafiken und effizienter Fehlerbehebung. Sie werden begeistert sein, was Sie mit MATLAB alles anstellen können.

This textbook teaches students techniques for the design of advanced digital systems using Field Programmable Gate Arrays (FPGAs). The authors focus on communication between FPGAs and peripheral devices (such as EEPROM, analog-to-digital converters, sensors, digital-to-analog converters, displays etc.) and in particular state machines and timed state machines for the implementation of serial communication protocols, such as UART, SPI, I2C, and display protocols, such as VGA, HDMI. VHDL is used as the programming language and all topics are covered in a structured, step-by-step manner.

Die aktualisierte 5. Auflage dieses Standardwerks beschreibt die, noch anhaltende, Entwicklung und Verbreitung der Generativen Fertigungstechnik über alle Branchen und viele Anwendergruppen hinweg. Leistungsfähige Production Printer arbeiten in der Industrie und Fabber, kleine, preiswerte und meist selbst zu bauende 3D-Drucker, erschließen die Generative Fertigung auch für Privatleute und an entlegenen Orten. Seriöse Journale und Tageszeitungen machen mit Druckern Erfolgsgeschichten auf. Drucker sind in aller Munde. Daneben wird die Technik sukzessive verbessert. Die Prozesse werden stabiler und vor allem reproduzierbar. Eine wirkliche Massenproduktion von Einzelteilen gelingt in einzelnen Branchen und beginnt sich durchzusetzen. Neu in der 5. Auflage sind: - Aktualisierungen: Firmen, Maschinen und Material; Anwendungsbeispiele - Erweiterungen: Fabbertechnologie, Do It Yourself Drucker EXTRA: E-Book inside Eines der Hauptprobleme beim Chipentwurf besteht darin, daß die Anzahl der zu bewältigenden Kombinationen der einzelnen Chipbausteine ins Unermeßliche steigt. Hier hat sich eine sehr fruchtbare Verbindung zu einem Kerngebiet der Theoretischen Informatik, dem Gebiet des Entwurfs von Datenstrukturen und effizienten Algorithmen, herstellen lassen: das Konzept der geordneten binären Entscheidungsgraphen, das in zahlreichen CAD-Projekten zu einer beträchtlichen Leistungssteigerung geführt hat. Die Autoren stellen die Grundlagen dieses interdisziplinären Forschungsgebiets dar und behandeln wichtige Anwendungen aus dem rechnergestützten Schaltkreisentwurf.

Wenn Sie programmieren können, beherrschen Sie bereits Techniken, um aus Daten Wissen zu extrahieren. Diese kompakte Einführung in die Statistik zeigt Ihnen, wie Sie rechnergestützt, anstatt auf mathematischem Weg Datenanalysen mit Python durchführen können.

Praktischer Programmier-Workshop statt grauer Theorie: Das Buch führt Sie anhand eines durchgängigen Fallbeispiels durch eine vollständige Datenanalyse -- von der Datensammlung über die Berechnung statistischer Kennwerte und Identifikation von Mustern bis hin zum Testen statistischer Hypothesen. Gleichzeitig werden Sie mit statistischen Verteilungen, den Regeln der Wahrscheinlichkeitsrechnung, Visualisierungsmöglichkeiten und vielen anderen Arbeitstechniken und Konzepten vertraut gemacht. Statistik-Konzepte zum Ausprobieren: Entwickeln Sie über das Schreiben und Testen von Code ein Verständnis für die Grundlagen von Wahrscheinlichkeitsrechnung und Statistik: Überprüfen Sie das Verhalten statistischer Merkmale durch Zufallsexperimente, zum Beispiel indem Sie Stichproben aus unterschiedlichen Verteilungen ziehen. Nutzen Sie Simulationen, um Konzepte zu verstehen, die auf mathematischem Weg nur schwer zugänglich sind. Lernen Sie etwas über Themen, die in Einführungen üblicherweise nicht vermittelt werden, beispielsweise über die Bayessche Schätzung. Nutzen

Sie Python zur Bereinigung und Aufbereitung von Rohdaten aus nahezu beliebigen Quellen. Beantworten Sie mit den Mitteln der Inferenzstatistik Fragestellungen zu realen Daten.

In dieser - lang erwarteten - Überarbeitung zur Version 2.0 der umfassenden Einführung in UML bieten die Entwickler der Sprache - Grady Brooch, James Rumbaugh, Ivar Jacobsen - eine Einführung, die sich mit den Kernpunkten befasst. Ausgehend von einer Übersicht über UML wird die Sprache anhand der Vorstellung bestimmter Konzepte und Schreibweisen in jedem Kapitel Schritt für Schritt erläutert. Das Buch sorgt einerseits für einen umfassenden Überblick über alle Diagrammtypen sowie Elemente von UML in der zweiten Version und stellt andererseits den nötigen Praxisbezug her, um UML 2.0 effektiv für eigene Projekte einzusetzen. Die tief greifenden Erläuterungen und die an Beispielen orientierte Herangehensweise der Autoren, sorgen für ein schnelles Verständnis des komplexen Themas.

Der Stil dieses Buches - leicht verständlich und mit großzügigem Layout - hat schon Tausende von Lesern begeistert. Nach "HTML & CSS" und "JavaScript & jQuery" erscheint jetzt "PHP & MySQL" von Jon Duckett. Lernen Sie, Websites zu erstellen, die leicht aktualisiert werden können und trotzdem jedem Benutzer andere Inhalte zeigen. Die Techniken in diesem Buch sind für alle Arten von Websites nützlich: Online-Shops, Kataloge, Blogs, soziale Netzwerke, Suchmaschinen und viele mehr. PHP- und MySQL-Kenntnisse sind im Übrigen auch bei der Verwendung von Content-Management-Systemen wie WordPress, Magento, Drupal und Joomla sehr nützlich.

Principles of Verifiable RTL Design: A Functional Coding Style Supporting Verification Processes in Verilog explains how you can write Verilog to describe chip designs at the RT-level in a manner that cooperates with verification processes. This cooperation can return an order of magnitude improvement in performance and capacity from tools such as simulation and equivalence checkers. It reduces the labor costs of coverage and formal model checking by facilitating communication between the design engineer and the verification engineer. It also orients the RTL style to provide more useful results from the overall verification process. The intended audience for Principles of Verifiable RTL Design: A Functional Coding Style Supporting Verification Processes in Verilog is engineers and students who need an introduction to various design verification processes and a supporting functional Verilog RTL coding style. A second intended audience is engineers who have been through introductory training in Verilog and now want to develop good RTL writing practices for verification. A third audience is Verilog language instructors who are using a general text on Verilog as the course textbook but want to enrich their lectures with an emphasis on verification. A fourth audience is engineers with substantial Verilog experience who want to improve their Verilog practice to work better with RTL Verilog verification tools. A fifth audience is design consultants searching for proven verification-centric methodologies. A sixth audience is EDA verification tool implementers who want some suggestions about a minimal Verilog verification subset. Principles of Verifiable RTL Design: A Functional Coding Style Supporting Verification Processes in Verilog is based on the reality that comes from actual large-scale product design process and tool experience.

Die Hardwarebeschreibungssprache VHDL (Very High Speed Integrated Circuit Description Language) dient dem Entwurf der Hardwarekomponenten für komplexe Computer- und Consumer-Anwendungen. In diesem Lehrbuch wird, immer vor dem Hintergrund der Digitaltechnik, eine Einführung in Grundkonzepte aber auch detaillierter Einblick in die konkrete Synthese anhand von Beispielen gegeben. Inhaltliche Neuerungen der 6. Auflage: Durchgängige Verwendung des IEEE-Standards zur VHDL-Arithmetik Auf vielfachen Wunsch der Leser: Ergänzung um einen Abschnitt zum VHDL-Entwurf von Testbenches Ergänzung des Kapitels "FIR-Filter" um die Modellierung systolischer FIR- Filter Erweiterung um ein neues Kapitel zur VHDL Implementierung der numerischen Integration. Dieser Abschnitt ermöglicht die Hardware-Modellierung nichtlinearer Systeme, z.B. in der Regelungstechnik.

Die Entwurfsmethoden zur Digitaltechnik erleben seit einigen Jahren einen wesentlichen Paradigmenwechsel. Bisherige Methoden und Kenntnisse zum Digitaltechnikentwurf sind nicht mehr ausreichend. Industrie und Wissenschaft verlangen darüber hinaus die Fähigkeit zur Modellierung mit der Hardwarebeschreibungssprache VHDL. Das Konzept dieses Lehrbuchs erfüllt diese Anforderungen, indem die wesentlichen Sprachelemente von VHDL Schritt für Schritt parallel zu den Grundkenntnissen zum digitalen Schaltungsentwurf eingeführt werden. Der Leser ist nach dem Studium dieses Lehrbuchs in der Lage, einfache digitale Systeme zu verstehen und zu entwerfen, weil er zu allen Komponenten Funktion, Zeitverhalten sowie ein geeignetes VHDL-Entwurfsmuster zuordnen kann. Der ausgezeichnete didaktische Aufbau unterstützt dabei: Jedem Kapitel sind Lernziele vorangestellt; immer wieder werden grafische und tabellarische Übersichten sowie vertiefende Beispiele verwendet; eine Vielzahl von Übungsaufgaben mit Musterlösungen dient zur Lernkontrolle.

h2> Kommentare, Formatierung, Strukturierung Fehler-Handling und Unit-Tests Zahlreiche Fallstudien, Best Practices, Heuristiken und Code Smells Clean Code - Refactoring, Patterns, Testen und Techniken für sauberen Code

Aus dem Inhalt: Lernen Sie, guten Code von schlechtem zu unterscheiden Sauberen Code schreiben und schlechten Code in guten umwandeln Aussagekräftige Namen sowie gute Funktionen, Objekte und Klassen erstellen Code so formatieren, strukturieren und kommentieren, dass er bestmöglich lesbar ist Ein vollständiges Fehler-Handling implementieren, ohne die Logik des Codes zu verschleiern Unit-Tests schreiben und Ihren Code testgesteuert entwickeln Selbst schlechter Code kann funktionieren. Aber wenn der Code nicht sauber ist, kann er ein Entwicklungsunternehmen in die Knie zwingen. Jedes Jahr gehen unzählige Stunden und beträchtliche Ressourcen verloren, weil Code schlecht geschrieben ist. Aber das muss nicht sein. Mit Clean Code präsentiert Ihnen der bekannte Software-Experte Robert C. Martin ein revolutionäres Paradigma, mit dem er Ihnen aufzeigt, wie Sie guten Code schreiben und schlechten Code überarbeiten. Zusammen mit seinen Kollegen von Object Mentor destilliert er die besten Praktiken der agilen Entwicklung von sauberem Code zu einem einzigartigen Buch. So können Sie sich die Erfahrungswerte der Meister der Software-Entwicklung aneignen, die aus Ihnen einen besseren Programmierer machen werden – anhand konkreter Fallstudien, die im Buch detailliert durchgearbeitet werden. Sie werden in diesem Buch sehr viel Code lesen. Und Sie werden aufgefordert, darüber nachzudenken, was an diesem Code richtig und falsch ist. Noch wichtiger: Sie werden herausgefordert, Ihre professionellen Werte und Ihre Einstellung zu Ihrem Beruf zu überprüfen. Clean Code besteht aus drei Teilen: Der erste Teil beschreibt die Prinzipien, Patterns und Techniken, die zum Schreiben von sauberem Code benötigt werden. Der zweite Teil besteht aus mehreren, zunehmend komplexeren Fallstudien. An jeder Fallstudie wird aufgezeigt, wie Code gesäubert wird – wie eine mit Problemen behaftete Code-Basis in eine solide und effiziente Form umgewandelt wird. Der dritte Teil enthält den Ertrag und den Lohn der praktischen Arbeit: ein umfangreiches Kapitel mit Best Practices, Heuristiken und Code Smells, die bei der Erstellung der Fallstudien zusammengetragen wurden. Das Ergebnis ist eine Wissensbasis, die beschreibt, wie wir denken, wenn wir Code schreiben, lesen und säubern. Dieses

Buch ist ein Muss für alle Entwickler, Software-Ingenieure, Projektmanager, Team-Leiter oder Systemanalytiker, die daran interessiert sind, besseren Code zu produzieren. Über den Autor: Robert C. »Uncle Bob« Martin entwickelt seit 1970 professionell Software. Seit 1990 arbeitet er international als Software-Berater. Er ist Gründer und Vorsitzender von Object Mentor, Inc., einem Team erfahrener Berater, die Kunden auf der ganzen Welt bei der Programmierung in und mit C++, Java, C#, Ruby, OO, Design Patterns, UML sowie Agilen Methoden und eXtreme Programming helfen. Das Buch spannt den Bogen von den Grundlagen der Digitaltechnik über den Entwurf mit VHDL zu den wichtigsten Komponenten digitaler Systeme. Die 7. Auflage wurde grundlegend überarbeitet und aktualisiert. Folgende Themen werden diskutiert: • Digitale Grundelemente wie Logikgatter und Flip-Flops • Kombinatorische und sequentielle Schaltungen • Schaltungsentwurf und Simulation mit VHDL • Programmierbare Logikbausteine (CPLDs, FPGAs) • Halbleiterspeicher • AD-/DA-Umsetzer • Architektur von Mikroprozessoren • Mikrocontroller Zahlreiche Beispiele erleichtern das Verständnis. Übungsaufgaben mit Musterlösungen unterstützen die Lernkontrolle und stehen zu jedem Kapitel zur Verfügung.

Effective Coding with VHDL Principles and Best Practice MIT Press

Für einen erfolgreichen Hardware Entwurf sind nicht nur VHDL-Kenntnisse wichtig, sondern auch Kenntnisse der FPGA-Schaltungstechnik und der Design Tools. Das vorliegende Buch stellt die Zusammenhänge zwischen diesen wichtigen Themen dar und bietet eine zielgerichtete Einführung in den Entwurf von digitalen Schaltungen und Systemen mit FPGAs. Beginnend mit den Grundlagen von VHDL sowie der CMOS- und FPGA-Technologie, werden anschließend der synthesesgerechte Entwurf mit VHDL und die synchrone Schaltungstechnik auf dem FPGA behandelt. Darüber hinaus werden auch die wesentlichen Entwurfswerkzeuge, wie Logiksynthese oder die statische Timing-Analyse, erläutert. Abgerundet wird das Buch mit einem Kapitel über High-Level Synthese, welche eine Umsetzung von C/C++-Code in eine VHDL-Implementierung ermöglicht. Der Leser erhält anhand vieler Code-Beispiele einen praxisorientierten Zugang zum Hardware-Entwurf mit FPGAs. Zielgerichtete Einführung in den digitalen Schaltungsentwurf Alle notwendigen Kenntnisse für den rechnergestützten Hardwareentwurf Frank Kesel studierte Elektrotechnik an der Universität Karlsruhe und promovierte an der Universität Hannover. Er war zehn Jahre in der Industrie im digitalen ASIC-Design tätig. Er ist seit 1999 Professor an der Hochschule Pforzheim mit dem Spezialgebiet FPGA-Design.

Information has become one of the most valuable assets in the modern era. Within the last 5-10 years, the demand for multimedia applications has increased enormously. Like many other recent developments, the materialization of image and video encoding is due to the contribution from major areas like good network access, good amount of fast processors e.t.c. Many standardization procedures were carried out for the development of image and video coding. The

advancement of computer storage technology continues at a rapid pace as a means of reducing storage requirements of an image and video as most situation warrants. Thus, the science of digital video compression/coding has emerged. This storage capacity seems to be more impressive when it is realized that the intent is to deliver very high quality video to the end user with as few visible artifacts as possible. Current methods of video compression such as Moving Pictures Experts Group (MPEG) standard provide good performance in terms of retaining video quality while reducing the storage requirements. Many books are available for video coding fundamentals. This book is the research outcome of various Researchers and Professors who have contributed a might in this field. This book suits researchers doing their research in the area of video coding. The understanding of fundamentals of video coding is essential for the reader before reading this book. The book revolves around three different challenges namely (i) Coding strategies (coding efficiency and computational complexity), (ii) Video compression and (iii) Error resilience. The complete efficient video system depends upon source coding, proper inter and intra frame coding, emerging newer transform, quantization techniques and proper error concealment. The book gives the solution of all the challenges and is available in different sections.

Sie ist elegant, schlank, modern und flexibel: Die Rede ist von Scala, der neuen Programmiersprache für die Java Virtual Machine (JVM). Sie vereint die Vorzüge funktionaler und objektorientierter Programmierung, ist typischer als Java, lässt sich nahtlos in die Java-Welt integrieren – und eine in Scala entwickelte Anwendung benötigt oft nur einen Bruchteil der Codezeilen ihres Java-Pendants. Kein Wunder, dass immer mehr Firmen, deren große, geschäftskritische Anwendungen auf Java basieren, auf Scala umsteigen, um ihre Produktivität und die Skalierbarkeit ihrer Software zu erhöhen. Das wollen Sie auch? Dann lassen Sie sich von den Scala-Profis Dean Wampler und Alex Payne zeigen, wie es geht. Ihre Werkzeugkiste: Schon bevor Sie loslegen, sind Sie weiter, als Sie denken: Sie können Ihre Java-Programme weiter verwenden, Java-Bibliotheken nutzen, Java von Scala aus aufrufen und Scala von Java aus. Auch Ihre bevorzugten Entwicklungswerkzeuge wie NetBeans, IntelliJ IDEA oder Eclipse stehen Ihnen weiter zur Verfügung, dazu Kommandozeilen-Tools, Plugins für Editoren, Werkzeuge von Drittanbietern – und natürlich Ihre Programmiererfahrung. In Programmieren mit Scala erfahren Sie, wie Sie sich all das zunutze machen. Das Hybridmodell: Die Paradigmen "funktional" und "objektorientiert" sind keine Gegensätze, sondern ergänzen sich unter dem Scala-Dach zu einem sehr produktiven Ganzen. Nutzen Sie die Vorteile funktionaler Programmierung, wann immer sich das anbietet – und seien Sie so frei, auf die guten alten Seiteneffekte zu bauen, wenn Sie das für nötig halten. Futter für die Profis: Skalierbare Nebenläufigkeit mit Aktoren, Aufzucht und Pflege von XML mit Scala, Domainspezifische Sprachen, Tipps zum richtigen Anwendungsdesign – das sind nur ein paar der fortgeschrittenen Themen, in die Sie mit den beiden Autoren eintauchen. Danach sind Sie auch Profi im Programmieren mit Scala.

Electronic systems based on digital principles are becoming ubiquitous. A good design approach to these systems is essential and a top-down methodology is favoured. Such an approach is vastly simplified by the use of computer modeling to describe the systems. VHDL is a formal language which allows a designer to model the behaviours and structure of a digital circuit on a computer before implementation. "Digital System Design with VHDL" is intended both for students on Digital Design courses and practitioners who would like to integrate digital design and VHDL synthesis in the workplace. Its unique approach combines the principles of digital design with a guide to the use of VHDL. Synthesis issues are discussed and practical guidelines are provided for improving simulation accuracy and performance. Features: a practical perspective is obtained by the inclusion of real-life examples an emphasis on software engineering practices encourages clear coding and adequate documentation of the process demonstrates the effects of particular coding styles on synthesis and simulation efficiency covers the major VHDL standards includes an appendix with examples in Verilog A completely updated and expanded comprehensive treatment of VHDL and its applications to the design and simulation of real, industry-standard circuits. This comprehensive treatment of VHDL and its applications to the design and simulation of real, industry-standard circuits has been completely updated and expanded for the third edition. New features include all VHDL-2008 constructs, an extensive review of digital circuits, RTL analysis, and an unequalled collection of VHDL examples and exercises. The book focuses on the use of VHDL rather than solely on the language, with an emphasis on design examples and laboratory exercises. The third edition begins with a detailed review of digital circuits (combinatorial, sequential, state machines, and FPGAs), thus providing a self-contained single reference for the teaching of digital circuit design with VHDL. In its coverage of VHDL-2008, it makes a clear distinction between VHDL for synthesis and VHDL for simulation. The text offers complete VHDL codes in examples as well as simulation results and comments. The significantly expanded examples and exercises include many not previously published, with multiple physical demonstrations meant to inspire and motivate students. The book is suitable for undergraduate and graduate students in VHDL and digital circuit design, and can be used as a professional reference for VHDL practitioners. It can also serve as a text for digital VLSI in-house or academic courses.

Gegenstand dieses Lehrbuchs ist die Behandlung schwer lösbarer diskreter Optimierungsprobleme. Im ersten Teil werden schnelle Algorithmen vorgestellt, die solche Probleme näherungsweise lösen können. Der zweite Teil behandelt Komplexitätstheorie und Nichtapproximierbarkeit von Optimierungsproblemen. Das Lehrbuch enthält zudem zahlreiche Anwendungsbeispiele, Übungsaufgaben, Illustrationen und Abschnitte über Grundlagen wie etwa die Turingmaschine. Typische Programmieraufgaben kreativ lösen am Beispiel von C++ Von der Aufgabe zur Lösung – so gehen Sie vor Probleme analysieren und schrittweise bearbeiten Systematisches Vorgehen lernen und anwenden Aus dem Inhalt:

Strategien zur Problemlösung Eingabeverarbeitung Statusverfolgung Arrays Zeiger und dynamische Speicherverwaltung Klassen Rekursion Wiederverwendung von Code Rekursive und iterative Programmierung Denken wie ein Programmierer Die Herausforderung beim Programmieren besteht nicht im Erlernen der Syntax einer bestimmten Sprache, sondern in der Fähigkeit, auf kreative Art Probleme zu lösen. In diesem einzigartigen Buch widmet sich der Autor V. Anton Spraul genau jenen Fähigkeiten, die in normalen Lehrbüchern eher nicht behandelt werden: die Fähigkeit, wie ein Programmierer zu denken und Aufgaben zu lösen. In den einzelnen Kapiteln behandelt er jeweils verschiedene Programmierkonzepte wie beispielsweise Klassen, Zeiger und Rekursion, und fordert den Leser mit erweiterbaren Übungen zur praktischen Anwendung des Gelernten auf. Sie lernen unter anderem: Probleme in diskrete Einzelteile zerlegen, die sich leichter lösen lassen Funktionen, Klassen und Bibliotheken möglichst effizient nutzen und wiederholt verwenden die perfekte Datenstruktur für eine Aufgabenstellung auswählen anspruchsvollere Programmiertechniken wie Rekursion und dynamischen Speicher einsetzen Ihre Gedanken ordnen und Strategien entwickeln, um bestimmte Problemkategorien in Angriff zu nehmen Die Beispiele im Buch werden mit C++ gelöst, die dargestellten kreativen Problemlösungskonzepte gehen aber weit über die einzelnen Programmiersprachen und oft sogar über den Bereich der Informatik hinaus. Denn wie die fähigsten Programmierer wissen, handelt es sich beim Schreiben herausragender Quelltexte um kreative Kunst und der erste Schritt auf dem Weg zum eigenen Meisterwerk besteht darin, wie ein Programmierer zu denken. Über den Autor: V. Anton Spraul hat über 15 Jahre lang Vorlesungen über die Grundlagen der Programmierung und Informatik gehalten. In diesem Buch fasst er die von ihm dabei perfektionierten Verfahren zusammen. Er ist auch Autor von »Computer Science Made Simple«.

Mochtest du Elektronik-Grundwissen auf eine unterhaltsame und geschmeidige Weise lernen? Mit Make: Elektronik tauchst du sofort in die faszinierende Welt der Elektronik ein. Entdecke die Elektronik und verstehe ihre Gesetze durch beeindruckende Experimente: Zuerst baust du etwas zusammen, dann erst kommt die Theorie. Vom Einfachen zum Komplexen: Du beginnst mit einfachen Anwendungen und gehst dann zugig über zu immer komplexeren Projekten: vom einfachen Schaltkreis zum Integrierten Schaltkreis (IC), vom simplen Alarmsignal zum programmierbaren Mikrocontroller. Schritt-für-Schritt-Anleitungen und über 500 farbige Abbildungen und Fotos helfen dir dabei, Elektronik einzusetzen -- und zu verstehen.

Mit der deutschen Übersetzung zur fünften Auflage des amerikanischen Klassikers Computer Organization and Design - The Hardware/Software Interface ist das Standardwerk zur Rechnerorganisation wieder auf dem neusten Stand - David A. Patterson und John L. Hennessy gewähren die gewohnten Einblicke in das Zusammenwirken von Hard- und Software, Leistungseinschätzungen und zahlreicher Rechnerkonzepte in einer Tiefe, die zusammen mit klarer Didaktik

und einer eher lockeren Sprache den Erfolg dieses weltweit anerkannten Standardwerks begründen. Patterson und Hennessy achten darauf, nicht nur auf das "Wie" der dargestellten Konzepte, sondern auch auf ihr "Warum" einzugehen und zeigen damit Gründe für Veränderungen und neue Entwicklungen auf. Jedes der Kapitel steht für einen deutlich umrissenen Teilbereich der Rechnerorganisation und ist jeweils gleich aufgebaut: Eine Einleitung, gefolgt von immer tiefgreifenderen Grundkonzepten mit steigender Komplexität. Darauf eine aktuelle Fallstudie, "Fallstricke und Fehlschlüsse", Zusammenfassung und Schlussbetrachtung, historische Perspektiven und Literaturhinweise sowie Aufgaben. In der neuen Auflage sind die Inhalte in den Kapiteln 1-5 an vielen Stellen punktuell verbessert und aktualisiert, mit der Vorstellung neuerer Prozessoren worden, und der Kapitel 6... from Client to Cloud wurde stark überarbeitet. Umfangreiches Zusatzmaterial (Werkzeuge mit Tutorien etc.) steht Online zur Verfügung.

- Die bekannten Design Patterns der Gang of Four im konkreten Einsatz für die Entwicklung von Games - Zahlreiche weitere vom Autor entwickelte Patterns - Sequenzierungs-, Verhaltens-, Entkopplungs- und Optimierungsmuster Für viele Spieleprogrammierer stellt die Finalisierung ihres Spiels die größte Herausforderung dar. Viele Projekte verlaufen im Sande, weil Programmierer der Komplexität des eigenen Codes nicht gewachsen sind. Die im Buch beschriebenen Design Patterns nehmen genau dieses Problem in Angriff. Der Autor blickt auf jahrelange Erfahrung in der Entwicklung von weltweit erfolgreichen Games zurück und stellt erprobte Patterns vor, mit deren Hilfe Sie Ihren Code entwirren und optimieren können. Die Patterns sind in Form unabhängiger Fallbeispiele organisiert, so dass Sie sich nur mit den für Sie relevanten zu befassen brauchen und das Buch auch hervorragend zum Nachschlagen verwenden können. Sie erfahren, wie man eine stabile Game Loop schreibt, wie Spielobjekte mithilfe von Komponenten organisiert werden können und wie man den CPU-Cache nutzt, um die Performance zu verbessern. Außerdem werden Sie sich damit beschäftigen, wie Skript-Engines funktionieren, wie Sie Ihren Code mittels Quadrees und anderen räumlichen Aufteilungen optimieren und wie sich die klassischen Design Patterns in Spielen einsetzen lassen.

A guide to applying software design principles and coding practices to VHDL to improve the readability, maintainability, and quality of VHDL code. This book addresses an often-neglected aspect of the creation of VHDL designs. A VHDL description is also source code, and VHDL designers can use the best practices of software development to write high-quality code and to organize it in a design. This book presents this unique set of skills, teaching VHDL designers of all experience levels how to apply the best design principles and coding practices from the software world to the world of hardware. The concepts introduced here will help readers write code that is easier to understand and more likely to be correct, with improved readability, maintainability, and overall quality. After a brief review of VHDL, the book presents fundamental design principles for writing code, discussing such topics as design, quality, architecture, modularity,

abstraction, and hierarchy. Building on these concepts, the book then introduces and provides recommendations for each basic element of VHDL code, including statements, design units, types, data objects, and subprograms. The book covers naming data objects and functions, commenting the source code, and visually presenting the code on the screen. All recommendations are supported by detailed rationales. Finally, the book explores two uses of VHDL: synthesis and testbenches. It examines the key characteristics of code intended for synthesis (distinguishing it from code meant for simulation) and then demonstrates the design and implementation of testbenches with a series of examples that verify different kinds of models, including combinational, sequential, and FSM code. Examples from the book are also available on a companion website, enabling the reader to experiment with the complete source code.

Lernen Sie in diesem Buch mehr über Algorithmen und Datenstrukturen In diesem Lehrbuch werden Algorithmen und Datenstrukturen exakt aber auch anschaulich und nachvollziehbar vermittelt, denn Algorithmen sind heute allgegenwärtig und vielfältig. Sie sind Gegenstand intensiver Forschung und zählen zu den fundamentalen Konzepten der Informatik. Dieses Buch über Algorithmen und Datenstrukturen ist aus Vorlesungen für Studierende der Informatik sowie der Medien- und Wirtschaftsinformatik an der Technischen Hochschule Nürnberg entstanden. Die grundlegenden Themen werden in den Bachelorkursen behandelt. Fortgeschrittene Teile, wie zum Beispiel die probabilistischen Algorithmen, stammen dagegen aus Masterkursen. Der Inhalt des Werks im Überblick Im ersten Kapitel seines Buchs über Algorithmen und Datenstrukturen führt Knebl relevante Grundlagen und Designprinzipien für Algorithmen ein. Die anschließenden Kapitel 2 – 6 sind nach Problembereichen organisiert: Sortieren und Suchen (2), Hashverfahren (3), Bäume zur Speicherung von Daten und zur Datenkomprimierung (4), fundamentale Graphenalgorithmen, wie Tiefen- und Breitensuche und Anwendungen davon (5), die Berechnung von minimalen aufspannenden Bäumen und von kürzesten Wegen in gewichteten Graphen als auch die Lösung des Flussproblems in Netzwerken (6). Probabilistische Methoden sind grundlegend für einfache sowie effiziente Algorithmen und Datenstrukturen. Deshalb wird in jedem Kapitel dieses Buchs mindestens ein Problem mit einem probabilistischen Algorithmus gelöst. Die notwendigen mathematischen Grundlagen werden im ersten Kapitel sowie im Anhang entwickelt. Lösungen zu den zahlreichen Übungsaufgaben stehen Ihnen bequem zum Download bereit.

Der Arduino ist eine preiswerte und flexible Open-Source-Mikrocontroller- Plattform mit einer nahezu unbegrenzten Palette von Add-ons für die Ein- und Ausgänge - wie Sensoren, Displays, Aktoren und vielem mehr. In "Arduino-Workshops" erfahren Sie, wie diese Add-ons funktionieren und wie man sie in eigene Projekte integriert. Sie starten mit einem Überblick über das Arduino-System und erfahren dann rasch alles über die verschiedenen elektronischen Komponenten und Konzepte. Hands-on-Projekte im ganzen Buch vertiefen das Gelernte Schritt für Schritt und helfen

