

7. E.I.S.-Workshop am 7./8. November 1995 an der TU Chemnitz-Zwickau

E.I.S.-Labore – eine Siegener Chronik

Bernd Klose, Walter Lang, Klaus Quibeldey-Cirkel, Andreas W. Wieland, Hans Wojtkowiak
Universität Siegen, Technische Informatik, 57068 Siegen

e-mail: {lang|quibeldey}@sieis.informatik.uni-siegen.de

Der technische Fortschritt relativiert die alten und fordert zugleich neue Qualifikationsprofile: *System-on-Chip* und *Programming-in-the-Large* beschreiben die Komplexität heutiger Systementwürfe. Gefordert ist die akademische Ausbildung: Wie kann der industrielle Anspruch an den Ingenieur als Systementwerfer berücksichtigt werden? Wie läßt sich der technologische Sprung von der masken- zur feldprogrammierbaren Hardware methodisch und didaktisch aufbereiten? Im Zuge der Technologie wandeln sich Form und Inhalt der Entwurfslabore. Der Beitrag beschreibt die Labor-Chronik der Technischen Informatik in Siegen. Den besonderen Schwerpunkt legen wir auf das Ausbildungskonzept *Projektlabor* im Rahmen einer „Informatik-Systemtechnik“ [11,17].

1 E.I.S.: vom ASIC zum feldprogrammierbaren System

Seit dem Wintersemester 1982/83 – 2 Jahre nach Erscheinen der wegweisenden Einführung von Mead & Conway [4] – wird der VLSI-Entwurf an der Universität Siegen gelehrt. Folgt man Peter Lockemann, alle 5–6 Jahre die Lehrinhalte der Informatik zu revidieren [13], so deckt sich die Siegener Labor-Chronik in etwa mit seiner Empfehlung: Die konzeptionellen Überlegungen zum „Heute“ unseres Entwurfslabors *Datenverarbeitung* gehen zurück bis 1990, das „Morgen“ unseres Projektlabors zur *Informatik-Systemtechnik* beginnt im Wintersemester 1996/97. In den gleichlautenden Abschnitten beschreiben wir Organisation und Ablauf der Labore. Inhaltlich haben mehrere Faktoren die Chronik bestimmt:

- heterogene Werkzeugumgebungen: CAx-Szenario und Entwurfsautomatisierung [14]

- Entfernung vom Chip-Layout: Die frühen Phasen, Systemanalyse und Systemdesign, rücken in den Vordergrund [15]
- Fusion der Belange von Soft- und Hardware: Codesign feldprogrammierbarer Systeme [7]
- Entwerfen als Gruppenprozeß: Kooperation und Kommunikation kennzeichnen die industrielle Entwurfssituation [6]

Mit den technologischen Möglichkeiten wandeln sich auch die curricularen Anforderungen an die Ausbildung des Systementwerfers:

1.1 Neuer Lehrrahmen: „Informatik-Systemtechnik“

Moderne rechnergestützte Anlagen und Geräte sind *komplexe* und *heterogene* Gebilde. Sie sind komplex, da sie sich aus verteilten Systemen zusammensetzen, die wiederum verteilte Systeme sein können. Sie sind heterogen, da sie soft- und hardwaretechnische Komponenten umfassen, wie Betriebssysteme und Compiler, Datenbanken und Anwendersoftware, Rechner und Peripherie sowie Sensoren und Aktoren. Die Systemteile können geographisch verteilt oder lokal eng gekoppelt sein. Beispiele sind Leitsysteme für Gebäudekomplexe oder Systeme der Telekommunikation.

Für diese rechnergestützten Anlagen und Geräte wird der Begriff „Informatiksysteme“ vorgeschlagen, im Englischen CBS: Computer-Based Systems. Die IEEE-Ingenieurdisziplin, die sich mit dem Entwurf von Informatiksystemen befaßt, heißt „Informatik-Systemtechnik“, englisch ECBS: Engineering of Computer-Based Systems [17]. Bild 1 skizziert ihre Einordnung. Demnach ist die Informatik-Systemtechnik eine Teildisziplin der allgemeinen Systemtechnik. Sie integriert die Methoden, Werkzeuge und Produkte etablierter Ingenieurtechniken, um Informatiksysteme zu

entwerfen. Diese steuern Anlagen und Geräte oder sind in Geräten und Anlagen eingebettet (*embedded systems*).

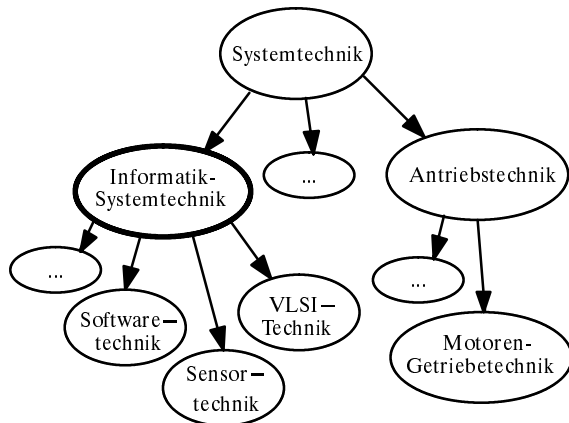


Bild 1: Einordnung der Informatik-Systemtechnik

Das hierfür spezifische Anforderungsprofil des Systementwerfers erläutern wir an anderer Stelle im Tagungsband [11]. Dort skizzieren wir die Didaktik des zukünftigen *Studienmodells* „Informatik-Systemtechnik“. Im folgenden werden die *praxisbezogenen* Leitbilder vom „flexiblen Entwerfen“ und der „Team-Kompetenz“ hervorgehoben:

1.2 Neue Leitbilder für Entwurfslabore

Flexibles Entwerfen: Redesigns sind prinzipiell unvermeidbar, denn innovative komplexe Entwürfe folgen dem Prinzip von Versuch und Irrtum. Redesigns sind aber kostspielig, wenn *masken-programmierbare* Schaltungen (ASIC) eingesetzt werden. *Feld-programmierbare* Schaltungen (FPGA) schaffen hier Abhilfe: Sie sind das Hardware-Pendant zum ausführbaren Zielcode eines Rechnerprogramms. Kommen weitere „weiche“ Hardware-Strukturen hinzu, wie programmierbare Analogschaltungen und Verbindungselemente (Sammelkürzel: FPD = Field-Programmable Device), so steht ein *durchgängig* feldprogrammierbares Zielmedium zur Verfügung. Flexibles Entwerfen ist dann von der Systemebene bis zur Hardware-Implementierung möglich: „Rapid Prototyping“ analog zur Software-Technik.

Komplexe FPGA-Bausteine ermöglichen weit mehr als wiederprogrammierbare Hardware: Das feldprogrammierbare elektronische System vereint die Laufzeiteffizienz der Hardware mit der Gestaltungsflexibilität der Software. Die Schaltungsauslegung dieser Systeme erfolgt *nach* der Auslieferung – *online* beim Kunden. Umbau oder Umrüstung aufgrund geänderter Kundenanforderungen sind *passé*.

Bild 2 zeigt den Trend im FPGA-Einsatz: vom TTL-Ersatz (Glue Logic) über dedizierte I/O-Prozessoren und Von-Neumann-Coprozessoren zum speicherorientierten System ohne Schaltungsfixierung [5].

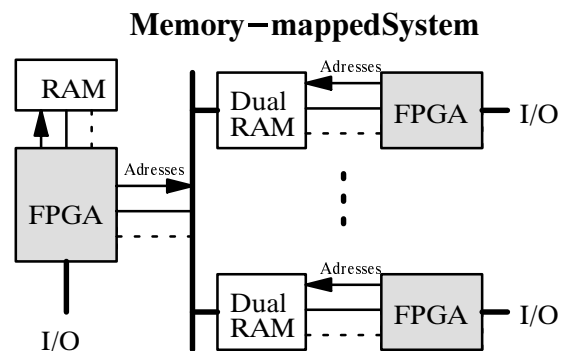
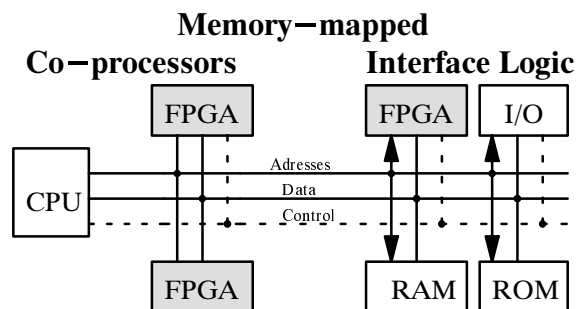
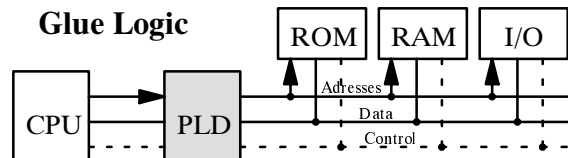


Bild 2: Vom PLD zum feldprogrammierbaren, speicherorientierten System

Unser Projektlabor reduziert diese Vision auf die Problemklasse der Steuerungssysteme mittlerer Komplexität. Bild 3 zeigt die Entwurfsumgebung;

unser Poster zum Beitrag (siehe Anhang) illustriert die Entwurfskomponenten (OOAD/OOP = Analyse, Design und Programmierung nach dem Objekt-Paradigma [15]; UPCS= User-Programmable Control System).

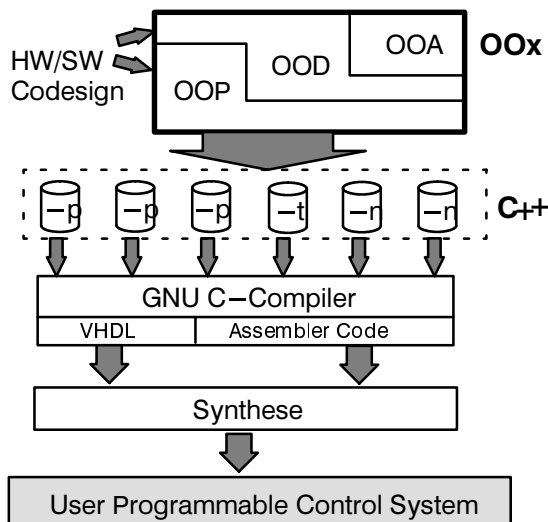


Bild 3: Werkzeug-Szenario der Projektlabors

Schon derartige Entwürfe machen die kooperative Vorgehensweise zwingend notwendig:

Team-Kompetenz: Neben den technischen Schnittstellen beeinflussen besonders die zwischenmenschlichen „Schnittstellen“ die Entwurfsqualität und -produktivität: Ein arbeitsteiliger Prozeß setzt kooperative und kommunikative Fähigkeiten voraus – *soft skills* [6]. Die soziale Kompetenz des Systementwerfers ist so wichtig wie seine fachliche. Kundenbefragungen, Projektbesprechungen, Delegation von Aufgaben, Auswahlgespräche mit Dienstleistern und Zulieferern, Präsentation von Ergebnissen und andere „zwischenmenschliche Situationen“ im Systementwurf fordern die *nicht-technische* Kompetenz.

Eingedenk dieser Anforderungen an die Qualifizierung des Systementwerfers bieten wir im Wintersemester 1996/97 ein organisatorisch und inhaltlich neues Entwurfslabor an:

1.3 Neues Erfahrungsforum: „Projektlabor“

Prinzipiell umfaßt auch der FPD-basierte Systementwurf die Entwurfsphasen, -methoden und -werkzeuge des traditionellen ASIC-Entwurfs: Systembeschreibung (hier allerdings als *Cospezifikation* von Soft- und Hardware), Synthese, Simulation und Test. Die ökonomischen und didaktischen Vorteile des feldprogrammierbaren Informatiksystems überwiegen jedoch [16]:

- hoher wirtschaftlicher Labor-Durchsatz: In [3] wird von einem FPGA-Labor berichtet, das jährlich 450 Teilnehmer im FPGA-Systementwurf ausbildet, wobei die Anschaffungskosten für Rechner und Werkzeuge bei einer Amortisation von 3 Jahren unglaubliche 30 ECU (ca. DM 56) pro Teilnehmer betragen!
- Praxisrelevanz: Methoden und Werkzeuge sind Stand der Technik und werden industriell eingesetzt. Es werden keine ASIC-Spezialisten ausgebildet, sondern Generalisten des anwendungsorientierten Systementwurfs.
- Rapid Prototyping: Die Laborteilnehmer haben ihre Aha-Erlebnisse umgehend, das heißt ohne die langen Turn-around-Zeiten maskenprogrammierter Chips. Den eigenen Chip *zum Laufen* zu bringen, ist didaktisch eine sehr wichtige und für den Entwerfer höchst befriedigende Erfahrung. Analoges gilt für den Validierungszyklus zwischen dem Kunden und dem Entwerfer: Frühe ausführbare Demo-Entwürfe schaffen Klarheit in den Anforderungen und Zuversicht in die Machbarkeit des Kundenauftrags.
- Portable Entwurfsumgebungen: Die Kooperation mit regionalen Ingenieurbüros des Anlagenbaus und der Automatisierungstechnik basiert auf gemeinsamen Betriebsmitteln (Industrie-PC mit Standard-Entwurfswerkzeugen, keine Hochleistungsrechner mit eigenentwickelten VLSI-Werkzeugen).

Der entscheidende Vorteil für eine *praxisbezogene* Lehre jedoch ist die *Erfahrungswelt*, die ein Projektlabor seinen Teilnehmern bietet. Unter simulierten industriellen Rahmenbedingungen lernen die Studierenden gruppendynamische Konfliktsituationen auszuhalten und zu bewältigen. Komplexe Informatiksysteme verlangen den *koopera-*

tiven Entwurf. Die erfolgreiche UPCS-Programmierung ist stets das Ergebnis eines langen (2 Semester, siehe Bild 4), komplexen (4–6 Personen) Gruppenprozesses. Darüber hinaus lernen die Studierenden, wie man anwendungsspezifische Information beim Kunden (Ingenieurbüro) und

aus der technischen Literatur rationell gewinnt. Regelmäßige Projektbesprechungen (Reviews), Präsentationen beim Kunden, im Seminar, Labor und schließlich im Kolloquium sind integraler Bestandteil des Projektlabors.

Vorlesung: 4 SWS Seminarübungen: 2 SWS Projektgruppen: 300 Stunden/Person

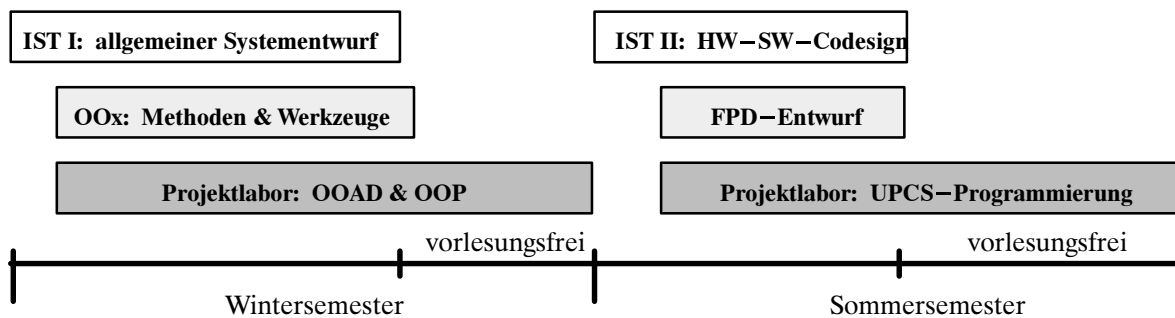


Bild 4: Zeitplan der Veranstaltung „Informatik-Systemtechnik – IST“

Das Abschlußdokument ist ein Projektbericht – formal das Äquivalent zur Studienarbeit. Für den Laborabsolventen ist dieses Äquivalent aber nicht nur eine Diplomvorleistung unter vielen: Aus der Warte des Bewerbungsempfängers dokumentiert der Projektbericht Teamfähigkeit, System- und Methodenkompetenz [11].

2 Gestern: Handentwurf

Der Beginn der Lehrveranstaltungen zum VLSI-Entwurf an der Universität Siegen fällt in das Wintersemester 1982/83. Zu diesem Zeitpunkt hatten sich Mitarbeiter der Fachgruppe für Technische Informatik mit Instituten anderer Universitäten zusammengeschlossen, um in einer gemeinsamen Anstrengung die hohen Kosten für eine Prototypenfertigung von Gate Arrays bei der Firma AEG zu tragen. Diese Aktivitäten wurden zu Beginn des Jahres 1983 in das E.I.S.-Projekt mit aufgenommen, an dem sich insgesamt 14 Hochschulen mit Unterstützung von Bund und Industrie und unter Federführung der GMD beteiligten. Ziel dieses Projekts war es, den VLSI-Entwurf zum Bestandteil der Ingenieurausbildung werden zu lassen, da die Industrie die mangelnden Fähigkeiten von Jungingenieuren in diesem Bereich beklagte.

Das E.I.S.-Labor an der Universität Siegen begann mit primitivsten Hilfsmitteln. Zu Anfang standen weder Logik- noch Layout-Synthesewerkzeuge zur Verfügung, so daß zunächst der Logikentwurf mit Papier und Bleistift durchgeführt werden mußte, um anschließend das Layout der Metallmaske mit Buntstiften auf einem großen Unterlegungsplot eines Gate Arrays UA4 der Firma AMI durchzuführen (Gatterkomplexität ca. 770 NAND-Äquivalente). Aus diesem gezeichneten Plot wurden dann die Koordinaten der Leiterbahnen berechnet und mit einem einfachen Texteditor die CIF-Datei erzeugt. Dieses naturgemäß sehr fehleranfällige Verfahren wurde durch die Einführung der (in der Fachgruppe entstandenen) Programme CIFEDIT, CIFANALYZE und CIFDISPLAY etwas vereinfacht. Ein interaktiver Entwurf (sowohl von Schaltbildern als auch von Layouts) mit einem graphischen Editor wurde jedoch erst mit dem (ebenfalls selbstgeschriebenen) ICE-Editor (Integrated Circuit Editor) im Wintersemester 1985/86 möglich.

Das Logiksynthese-Werkzeug LOG/IC wurde im Wintersemester 1986/87 eingeführt. Trotzdem bereitete die Validierung der entworfenen Schaltungen nach wie vor Probleme. Obwohl von Anfang an der Simulator DISIM zur Verfügung stand, war eine funktionale Simulation mühsam, da die DI-

SIM-Netzlisten noch von Hand geschrieben werden mußten. Das Layout konnte nur validiert werden, indem daraus bei der GMD eine Netzliste extrahiert wurde, die dann mit der von Hand entworfenen verglichen wurde.

Erst der ICE-Editor in Verbindung mit den Programmen NESSI und VERA ermöglichte eine Validierung der entworfenen Schaltungen in unserer Fachgruppe. NESSI konnte aus dem Schaltbild der Schaltung in CIF-Format eine DISIM-Netzliste extrahieren, die dann zur Logik- und Fehlersimulation verwendet wurde. VERA extrahierte aus dem fertigen Layout ebenfalls eine DISIM-Netzliste, die aber zusätzlich alle Laufzeitdaten enthielt, so daß auch eine Timing-Simulation durchgeführt werden konnte. Zusätzlich führte VERA einen Design Rule Check aus.

Das schwierigste Problem aber war der Test der gefertigten Schaltungen. Dieser konnte erst nach der Anschaffung eines Logikanalysators DAS 9100 der Firma Tektronix und der Erstellung des Programms DVTS ohne großen Zeitaufwand durchgeführt werden. Das Programm DVTS erzeugte die Eingabedaten für den Logikanalysator direkt aus den Entwurfsdaten. Unter Einbeziehung dieses Programms entstand bei der Firma Tektronix schließlich ein Chiptester. Dieser Tester wurde im Jahre 1990 durch einen kommerziellen Tester (LV500) von Tektronix ersetzt, um eine praxisnahe Ausbildung sicherzustellen.

Das E.I.S.-Labors wurde im Jahr 1991 durch Umstrukturierungen innerhalb des Fachbereichs Elektrotechnik und Informatik an eine andere Fachgruppe abgegeben. Dort wird seitdem mit kommerziellen Entwurfssystemen der Entwurf von integrierten Schaltungen gelehrt. In unserer Fachgruppe wurde daraufhin das Praktikum *Datenverarbeitung* zum digitalen Schaltungsentwurf mit feldprogrammierbaren Bausteinen aufgebaut.

3 Heute: FPGA-Entwurf

Der Schwerpunkt der Veranstaltung *Datenverarbeitung* liegt im Entwurf von Schaltnetzen und Schaltwerken. Als Zieltechnologien für dieses Praktikum kommen sowohl einfache (GAL) als

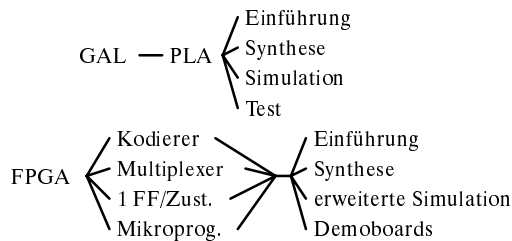
auch komplexe feldprogrammierbare Bausteine (Xilinx FPGA) zum Einsatz. Diese sind einerseits kostengünstig und leicht zu beschaffen und darüber hinaus durch ihre Wiederprogrammierbarkeit sehr änderungsfreundlich. Zudem erlauben sie auch den Entwurf komplexer Schaltungen: Mit modernen FPGA kann man mittlerweile Schaltungen in einer Komplexität von bis zu 25000 Gatteräquivalenten realisieren (Stand: März 1995).

Die Hauptaufgabe des Praktikums ist, die Lehrinhalte der Vorlesung *Datenverarbeitung* zu vertiefen. Im einzelnen sind dies die Boolesche Algebra, die Optimierung logischer Funktionen, die Schaltungsanalyse und der Entwurf von Schaltnetzen und Schaltwerken mit vorgefertigten Bausteinen (MUX, Register, ALU etc.).

Das Praktikum beginnt mit einer Einführung in die Synthese digitaler Schaltungen mit industriellen Entwicklungswerkzeugen. Es wird sowohl die sprachorientierte Synthese mit ABEL-HDL (der Firma DataI/O) als auch eine graphische Synthese (auf Register-Transfer- und Gatterebene) mit ViewDraw (der Firma ViewLogic) gelehrt.

Die Validierung der synthetisierten Schaltungen erfolgt durch Simulation. Die im Labor verwendeten Bausteine werden vom Hersteller bereits zu 100 % elektrisch getestet geliefert; deshalb wird keine Fehlersimulation durchgeführt, sondern das Hauptaugenmerk auf die Logik- und Timing-Simulation gelegt. Insbesondere wird auf die Methoden zur Erzeugung der Simulationsdaten großer Wert gelegt.

Im Anschluß an diesen eher theoretischen Teil folgt die erste Entwurfsaufgabe: Nach einem kurzen Überblick über PLD-Architekturen wird ein praktischer Entwurf mit einem GAL durchgeführt. Die so entworfene Schaltung wird dann getestet, wobei ausführlich auf den Zusammenhang von Simulation und Test und die verwendete Testerarchitektur eingegangen wird. Der Test wird auf dem fachgruppeneigenen Tester LV 500 der Firma Tektronix durchgeführt, wobei auch Testparameter (Betriebsspannung, *Setup*- und *Hold*-Zeiten etc.) dynamisch verändert werden (*Schmoo-Plot*).



Nach der Vermittlung aller Vorkenntnisse zur Schaltungssynthese und Simulation erfolgt ein komplexerer FPGA-Entwurf. Dazu wird zunächst die Architektur der Xilinx-FPGA und der Entwurfsablauf mit diesen Bausteinen detailliert dargestellt. Der Unterschied zwischen der Gatterentwurfsebene und den Logikblöcken eines Xilinx-FPGA wird durch einen kleinen Handentwurf verdeutlicht. Den Abschluß bildet dann ein komplexer Entwurf mit diesen Bausteinen.

Diese Form des Labors findet turnusmäßig im Sommersemester statt. Dabei werden 36 Studierende in Dreiergruppen ausgebildet. Legt man eine Amortisation der Betriebsmittel (3 PC plus Entwurfssoftware) von 3 Jahren zugrunde, kostet das Labor pro Teilnehmer 130 ECU (ca. DM 242). Zum Vergleich: ein *Full-Custom*-Entwurfslabor kostet bei gleicher Amortisationszeit 1200 ECU (ca. DM 2232) pro Teilnehmer (Angaben laut Siegener Fachgruppe für Rechnerstrukturen). Die Einführung von weiteren programmierbaren Bausteinen aus der Analog- und Verbindungstechnik ermöglichen den Aufbau eines kompletten Informatiksystems.

4 Morgen: FPD-Entwurf

Das zukünftige Projektlabor *Informatik-Systemtechnik* wird feldprogrammierbare Bausteine – FPD – in ihrer gesamten Bandbreite vorstellen:

- EPAC – Electrically Programmable Analog Circuit,
- FPID – Field Programmable Interconnect Device und

- FPGA – Field Programmable Gate Array.

Integraler Bestandteil der Laborform ist die Vermittlung von praktischen Kenntnissen auf den Gebieten Mixed-Signal-Design und Hardware-Software-Codesign mit diesen Bausteinen. Ziel ist das feldprogrammierbare Steuerungssystem [10], das in der Lage ist, die analogen und digitalen Signale von Sensoren und Aktoren zu verarbeiten.

Das Ergebnis eines solchen FPD-basierten Systementwurfs wird mit Hilfe eines vorgefertigten, benutzer-programmierbaren Steuerungssystems (User-Programmable Control System – UPCS) realisiert. Dieses UPCS besteht aus folgenden Komponenten (Bild 5):

SRISC-FPGA Prozessorkern mit unterschiedlicher Leistungsfähigkeit und variabler Speicherschnittstelle mit Hilfe eines

Memory-FPID zur Verteilung der SRISC-FPGA Speicherschnittstelle auf die einzelnen

RAM-Bausteine, die den Befehls- und Datenspeicher des Prozessors darstellen. Der Prozessor verfügt über zwei weitere Schnittstellen, nämlich die zwei *Ports* zu den seriellen- und parallelen Schnittstellen (*S1*, *S2*, *P*) und dem

Port-FPID zur Verteilung der *FPGA Ports* auf die I/O-Kopplung mit den *Sensoren* und *Aktoren*. Als Hardware realisierte Teile des Quellcodes werden über

Dual-Ported RAM-Bausteine (Realisierung der temporären Register und Kopplung mit dem Prozessorkern) an die

FPGA-Coprozessoren angeschlossen. Die Ports dieser *FPGA*-Coprozessoren werden ebenfalls über das *Port-FPID* mit den *Sensoren* und *Aktoren* verbunden.

Die UPCS-Programmierung ist, wie bereits in Abschnitt 1.3 dargestellt, wesentlicher Bestandteil des neuen Projektlabors "Informatik-Systemtechnik".

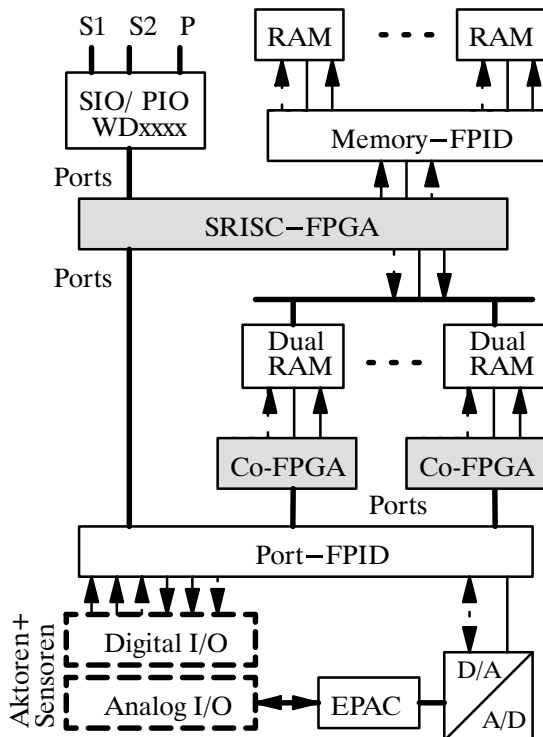


Bild 5: Blockschaltbild des UPCS

In dem in Bild 3 dargestellten Werkzeug-Szenario folgt nach der objektorientierten Systemanalyse (OOA) und des darauffolgenden Designs (OOD) eine Systembeschreibung in der Hochsprache C++ (OOP). Die einzelnen Module sind mit Schaltern zur Steuerung des C-Compilers versehen (Bild 6):

- Module, die mit dem Schalter `-n` (wie normal, d. h. laufzeit-unkritisch) kompiliert werden, ergeben ein ausführbares Programm für den SRISC-Prozessor. Dabei können die Systemdesigner aus der SRISC-Library aus einer Anzahl vorentworfener SRISC-Prozessoren den für die Anwendung geeignetsten auswählen.

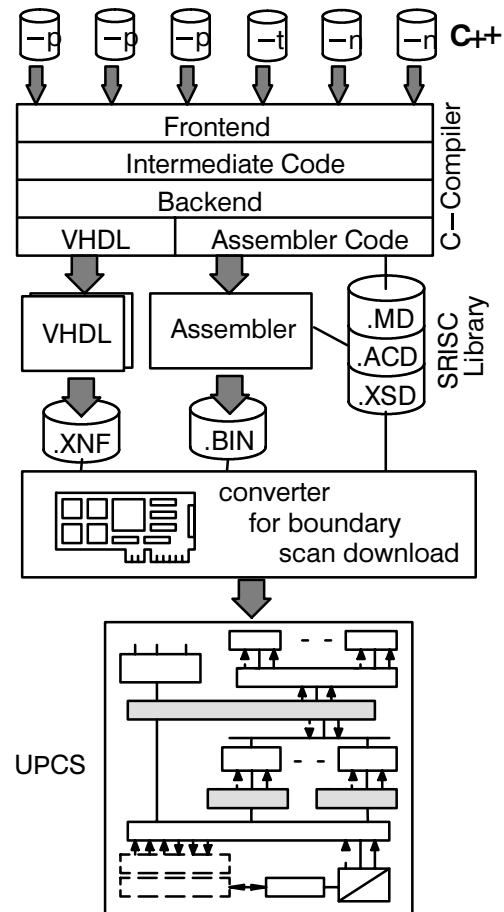


Bild 6: Der UPCS-Entwurf

Die Maschinenbeschreibung des gewählten Prozessors ist als `.MD`-Datei (Machine Description) in der SRISC-Library enthalten. Der Compiler benutzt diese, um für den speziellen Prozessor ein Assemblerprogramm zu erzeugen. Dieses wird unter Berücksichtigung der Assemblerbeschreibung des gewählten Prozessors (`.ACD`-Datei – Assembler Code Description) zu einem ausführbaren Programm (`.BIN`) assembliert. Die Hardware-Beschreibung des Prozessor-Operations- und Steuerwerks liegt als `.XSD`-Datei (Xilinx SRISC Description) ebenfalls in der SRISC-Library vor.

- Module, die mit dem Schalter `-p` (wie parallel, d. h. nebenläufig) kompiliert werden, erzeugen eine verhaltensorientierte Beschreibung (VHDL) eines FPGA-Coprozessors [8]. Die-

se wird mit einem VHDL-Compiler in ein Netzlistenformat der Zieltechnologie (in unserem Fall: .XNF – Xilinx Netlist Format) übersetzt. Diese Coprozessoren können beispielsweise dazu verwendet werden, um Teile der zu entwerfenden Steuerung mehr oder weniger unabhängig vom SRISC-Prozessor zu realisieren.

- Module, die mit dem Schalter $-t$ (wie time-critical) compiliert werden, berücksichtigen durch den Entwerfer vorgegebene Zeitbedingungen. Abhängig davon werden diese Module entweder zu Assemblercode für den gewählten SRISC-Prozessor compiliert oder in verhaltenensorientierten VHDL-Code umgesetzt.

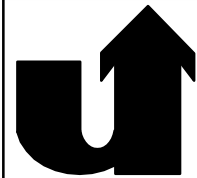
Nach der Assemblierung aller Programme und der Synthese der Coprozessoren wird die Programmierinformation für alle Bausteine generiert. Abschließend werden die Programmierdaten aller FPD-Bausteine und das ausführbare SRISC-Programm in eine Boundary-Scan-Sequenz

übersetzt [1], da diese Daten über den Boundary-Scan-Testbus in das UPCS geladen werden. Der Boundary-Scan-Testbus [9] dient hierbei als Transportmedium, da er von allen FPD-Bausteinen unterstützt wird und mit Hilfe des EXTEST-Befehls das Beschreiben der RAM-Bausteine ermöglicht.

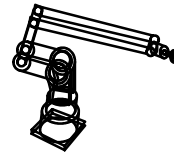
Die analogen Komponenten des UPCS werden zur Zeit noch unabhängig von dem beschriebenen Werkzeug-Szenario entworfen. Die Integration aller beschriebenen Komponenten (Bild 3) in den Entwurf eines Informatiksystems ist Forschungsgegenstand unserer Fachgruppe. Die Werkzeuge des Projektlabors werden ständig dem Forschungsstand angepaßt. In dieser Weise verzahnt unser Werkzeug-Szenario die Forschung mit der Lehre zum Thema Informatik-Systemtechnik.

Literatur

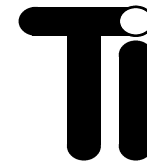
- [1] Bauer, C.: Kopplung des PC-ISA-Bus mit dem Boundary-Scan-Bus. Interner Bericht, Fachgruppe Technische Informatik der Universität Siegen, Siegen, 1995
- [2] Berten, A.: Objektorientierte Methoden zur Systemmodellierung und -spezifikation. Interner Bericht zum Hardware/Software Co-design-Seminar, Wintersemester 1994/95, Universität Siegen
- [3] Boemo, E. et al.: Field-Programmable Logic in Education: a Case Study. In: [12], S. 452-457
- [4] Conway, L.; Mead, C.: Introduction to VLSI Systems. Addison-Wesley Publishing Company, Reading, Mass., 1980
- [5] Dunlop, J.; Lysaght, P.: Dynamic Reconfiguration of FPGAs. In: [12], S. 82-94
- [6] Frese, M.; Hesse, W.: Zur Arbeitssituation in der Software-Entwicklung: Resümee einer empirischen Untersuchung. In: Informatik Forschung und Entwicklung, Jg. 9, 1994, S. 179-191
- [7] Glunz, W.: Hardware-Entwurf auf abstrakten Ebenen unter Verwendung von Methoden aus dem Software-Entwurf. Dissertation, Paderborn, 1994
- [8] Jantsch, A.; Ellervee, P.; Öberg, J.; Hemani, A.; Tenhunen, H.: Hardware-Software Partitioning and Minimizing Memory Interface Traffic. In: Proceedings of EURO-DAC-94, 1994
- [9] Klose, B.: Boundary-Scan-Architekturen. Interner Bericht, Fachgruppe für Technische Informatik der Universität Siegen, Siegen, 1993
- [10] Klose, B.; Lang, W.; Stocksiefen, F.; Wieland, A.: Anwenderprogrammierbare Steuerungssysteme. In: Tagungsband 2. GI/ITG Workshop „Anwenderprogrammierbare Schaltungen“, 22.-23. Juni 1995, Karlsruhe, FZI-Publikation 2/95
- [11] Lang, W.; Quibeldey-Cirkel, K.; Wojtkowiak, H.: 7 Leitbilder für die Lehre des Systementwurfs: ein Studienmodell der Informatik-Systemtechnik. In: Tagungsband zum 7. E.I.S.-Workshop an der TU Chemnitz-Zwickau, 1995
- [12] Luk, W.; Moore, W. R. (Hrsg.): More FPGAs. Abingdon EE&CS Books, Oxford GB, 1994
- [13] N. N.: Interview: Professor Peter Lockemann und Dr. Wilhelm Denz zur Situation der Informatik-Ausbildung: Den stabilen Arbeitsplatz wird es nicht mehr geben. Informatik-Spektrum, Jg. 18, H. 2, 1995, S. 111-113
- [14] Quibeldey-Cirkel, K.: CAD-Frameworks: Die Probleme jenseits der Entwurfswerkzeuge. In: Mikroelektronik, Jg. 7, H. 2, 1993, S. 72-76
- [15] Quibeldey-Cirkel, K.: Das Objekt-Paradigma in der Informatik. Teubner-Verlag, Stuttgart, 1994
- [16] Schubert, E. et al.: The Use of FPGAs for Educational Purposes in VLSI Microprocessor Design. In: [12], S. 458-465
- [17] Schweizer, G.; Thomé, B.: Informatik-Systemtechnik (ECBS): Gedanken zu einer Disziplin. Informatik-Spektrum, Jg. 16, 1993, S. 215-221



Universität
Siegen



Technische
Informatik



Anhang: Poster zum Beitrag

E.I.S.-Labor von morgen: Projektlabor zur Informatik – Systemtechnik

