Rekonfigurierbare Prozessoren in der Bildverarbeitung

Hans-Ulrich Zühlke, Axel Kaiser

Fakultät für Mathematik und Informatik
Lehrstuhl für Rechnerarchitektur (Prof. Erhard)
Friedrich-Schiller-Universität Jena, 07740 Jena
email: hans-ulrich.zuehlke@uni-jena.de

Schneller als die Leistungen der Computer wachsen die Anforderungen, die durch die Anwender und ihre Algorithmen gestellt werden. Insbesondere in der Bildverarbeitung werden oft einfache Operationen auf große Datenmengen mit sehr unterschiedlichen Datenformaten angewendet. Klassische Prozessorarchitekturen sind für solche Aufgaben wenig effizient. Eine Alternative zur aufwendigen Skalierung herkömmlicher Rechnersysteme ist die Nutzung rekonfigurierbarer Hardware. Am Lehrstuhl entstand neben konzeptionellen Arbeiten zu rekonfigurierbaren Systemen ein physisches Prozessormodell, der DCP. Die Arbeit wird durch die DFG im Rahmen des SFB182, TP A1E, gefördert.

Keywords: Bildverarbeitung, CCM, Coprozessor, DCP, FPGA

1 Das Zusammenwirken von Architektur und Algorithmus

1.1 Universalprozessoren vs. Spezialprozessoren

Seit über 20 Jahren werden Universalprozessoren nach dem von Neumann Paradigma entwikkelt. Die Leistung dieser klassischen Universalprozessoren wächst mit dem technologischen Fortschritt bei der Herstellung und durch zahlreiche Detailverbesserungen. Prinzipbedingt werden diese Prozessoren durch drei Faktoren in ihrer Leistung negativ beeinflußt:

- Die Architektur der Universalprozessoren ist so optimiert, daß ein breites Spektrum von Algorithmen in hinreichend kurzer Zeit abgearbeitet wird. Alle direkt in Hardware verfügbaren Prozessorbefehle werden besonders schnell ausgeführt, eine Tatsache, die bei RISC- und mathematischen Coprozessoren genutzt wird. Die Integration immer neuer Befehle in den Prozessor hat jedoch technische und ökonomische Grenzen. Zusätzliche Befehle, z.B. für Operationen der Bildverarbeitung, verbrauchen für anteilig immer seltener genutzte Befehle immer größere Si-Ressourcen auf dem Prozessor. Sinnvoller ist die Nutzung einer veränderlichen (rekonfigurierbaren) Hardware, die jeweils zur Laufzeit eines Algorithmus eine angenähert optimale Architektur zur Verfügung stellt.
- Universalprozessoren arbeiten mit festen Datenformaten. Die wirklich genutzten Datenbreiten ergeben sich aus den abzubildenden Algorithmen. Bitorientierte Ressourcen können durch Skalierung in der Datenbreite optimal an die jeweilige Aufgabe angepaßt werden.
- Häufig wird versucht, die Leistung von Systemen durch Parallelisierung zu verbessern. Die gesamte Anwendung kann jedoch nicht schneller als der jeweils größte sequentielle Subalgorithmus ausgeführt werden (Amdahls Law [1]). Für die optimale Ausführung der gesamten Anwendung ist deshalb die Beschleunigung der sequentiellen Anteile besonders wichtig. Dies

gilt unter anderem für die schnelle Vorverarbeitung eines Bildes vor der Anwendung paralleler high-level-Operationen.

Die drei genannten Nachteile der Universalprozessoren sind durch rekonfigurierbare Architekturen vermeidbar. Sie gestatten die temporäre Bereitstellung optimierter Befehle mit variabler Datenbreite bei gleichzeitig hohem Datendurchsatz.

1.1.1 Anforderungen an eine Architektur für die Bildverarbeitung

Die Algorithmen in der Bildverarbeitung haben viele unterschiedliche Eigenschaften und daraus resultierend vielfältige Anforderungen an das System. Weitere Bedingungen ergeben sich aus den subjektive Gewohnheiten des Anwenders. Einige grundlegende Anforderungen an die Architektur lassen sich dennoch beschreiben:

- Bei der Bearbeitung von ganzen Videosequenzen sind große Datenmengen mit hoher Datenrate zu verarbeiten. Für die meisten Echtzeitanwendungen sind geringe Verzögerungen tolerierbar, solange sie die Interaktion mit dem System nicht beeinträchtigen. Die Bildwiederholrate für Video-Sequenzen sollte dagegen konstant 25 Hz betragen. Viele Algorithmen können nach Pipeline- oder Datenstromprinzip implementiert werden. Dies kommt der Implementierung in FPGA entgegen.
- In der Bildverarbeitung variiert die Bitbreite für die Repräsentation eines Bildpunktes zwischen einem Bit (Binärbild) und bis zu 24 Bit bei Echtfarbdarstellung.
- Besonders im Bereich der Bildvorverarbeitung sind einfache Operatoren auf große Datenmengen anzuwenden (z.B. Filteroperatoren zur Bildverbesserung).
- Für lokale Operatoren und das Speichern von Zwischenergebnissen sind lokale Speicherressourcen erforderlich, die möglichst in der Nähe der zugreifenden Verarbeitungseinheiten platziert sein sollten.

Diese Anforderungen wurden durch die Studie zum DIBA-Projekt [2] bestätigt. Für eine computerbasierte Befunderhebung in der Radiologie sind Einzelbilder mit hoher Bildqualität und -auflösung erforderlich (2048 x 2048 Pixel mit je acht Bit = 4 MByte Daten). Es werden nur Einzelbilder betrachtet. Zur Gewährleistung des wahlfreien Zugrif auf bis zu 200 Bilder/h (20 Patienten mit jeweils maximal 10 Aufnahmen) ist trotzdem eine hohe Datenrate erforderlich. Die Erfüllung dieser Anforderungen mit Systemen aus herkömmlichen Prozessoren ist aufwendig. Für diese und ähnliche Anwendungen bietet sich die Nutzung rekonfigurierbarer Hardware an.

2 Konfigurierbare Hardware

2.1 FPGA

Für die Anforderungen des Rapid Prototyping wurden in den letzten Jahren *F*ield*p*rogrammable *G*ate *A*rrays (FPGA) entwickelt. Diese Bauelemente sind RAM-basiert und arbeiten bitorientiert. Die Schaltkreise bestehen im wesentlichen aus einer Matrix von einfachen Logik-Blöcken (CLB) und Routingressourcen. Ihre endgültige Funktionalität erhalten die FPGA durch das Konfigurieren innerhalb der arbeitenden Schaltung. Bei diesem Vorgang werden alle zuvor generierten Informationen zur Funktionalität der CLB's und ihre Verknüpfung untereinander in den FPGA übertragen. Das Schreiben erfolgt wie bei einem normalen RAM-Baustein, weshalb der Konfigurationsprozeß sehr schnell erfolgt. Nach dem Konfigurieren arbeitet der FPGA in etwa mit der Geschwindigkeit reiner Hardwarelösungen. Der maximale Arbeitstakt hängt von der jeweiligen Konfiguration ab. Rein synchrone Designs arbeiten am schnellsten.

Deshalb eignet sich auch das in der Bildverarbeitung häufig verwendete Pipelineprinzip besonders gut.

2.2 VHDL und der Design-Prozeß

Die Beschreibung der Funktionalität eines FPGA, aus der die entsprechende Konfiguration generiert wird, kann auf verschiedenen Abstraktionsebenen erfolgen. Eine sehr komfortable Variante ist die Beschreibung in der „Hardware-Hochsprache“ VHDL (*V*ery High Speed Integrated Circuit *H*ardware *D*escription *L*anguage). VHDL wurde 1983 vom amerikanischen Department of Defense initiiert und ist seit Ende 1987 als IEEE Standard 1076 genormt [4]. Inzwischen ist VHDL die defakto Standard-Hardware-Beschreibungssprache, die sich auch sehr gut für die Beschreibung von FPGA-Konfigurationen eignet.

Alle wesentlichen Sprachkonstrukte, die von „C“ oder „ADA“ bekannt sind, wurden auch in VHDL realisiert. Zusätzlich wurde unter anderem die Beschreibung von Timing-Bedingungen implementiert, die für Simulation von Vorgängen und Systemen notwendig sind.

Beschreibungen können hierarchisch gegliedert werden. Es gibt zwei wesentliche Möglichkeiten der Abstraktion, die meist kombiniert angewendet werden. Jede Design-Einheit (Entity) kann durch ihr Verhalten oder ihre Struktur beschrieben werden. Das Verhalten wird analog herkömmlichen Programmiersprachen algorithmisch beschrieben. Bei der strukturalen Beschreibung werden dagegen die Verknüpfungen von hierarchisch untergeordneten Elementen festgelegt. Die komplette Beschreibung eines FPGA wird dann durch entsprechende Entwicklungstools in immer kleinere Struktureinheiten gegliedert, bis diese unter Anwendung von schaltkreisspezifischen Bibliotheken in die Zielarchitektur abgebildet werden können.

2.3 Voraussetzungen für die Nutzung konfigurierbarer Hardware

Grundsätzlich lohnt sich der Einsatz eines konfigurierbaren Prozessors nur, wenn der zusätzliche Aufwand für Datentransfer und Konfiguration kleiner ist als der erreichte Gewinn an Rechenleistung.

In diese Betrachtung wurde die Erstellung der erforderlichen Konfigurationen nicht einbezogen. Sie erfolgt vor der Laufzeit der Anwendung, ist jedoch mit relativ hohem Aufwand verbunden. Zur Zeit werden größere Konfigurationen noch immer manuell von erfahrenen Entwicklern erstellt. Methoden und Tools zur automatischen Generierung von Konfigurationen werden an verschiedenen Stellen intensiv erforscht (unter anderem [3]). Parallel zu der FPGA-Konfiguration muß in den meisten Fällen eine entsprechende Software zur Bedienung und Nutzung des FPGA erstellt werden bzw. vorhandene Programme müssen verändert werden.

3 Konzept und Realisierung des DCP

3.1 Konzept des DCP

Die Integration der konfigurierbaren Ressource in das Gesamtsystem kann sehr unterschiedlich erfolgen. Bei reinen *C*ustom *C*omputing *M*achines (CCM) wird die gesamte Anwendung auf eine große Ressource aus mehreren FPGA portiert. Herkömmliche Universalprozessoren werden nur für sekundäre Dienste, die Konfiguration oder als Terminal eingesetzt.

Die konfigurierbare Ressource kann aber auch die Funktion eines Coprozessors für einen lokalen oder mehrere Universalprozessoren (shared Coprozessing) übernehmen. Die Nutzung als Coprozessor hat für den Anwender mehrere Vorteile. Die Eigenschaften des Gesamtsystems und dessen Bedienung bleiben weitgehend erhalten. Die Migration von der Software- zur

Hardwarelösung kann schrittweise erfolgen. Aufrufe von Subalgorithmen oder Software-Bibliotheksfunktionen werden nur bei Bedarf durch Konfigurationen ersetzt. Lokale Coprozessoren können je nach Verknüpfung mit dem Universalprozessor und der Art der Speicheranbindung lose oder eng gekoppelt sein. Bei sehr enger Kopplung kann die Aufgabenteilung auf der Ebene einzelner Operationen erfolgen.

Mit relativ geringem Aufwand an Hardware können viele Applikationen deutlich beschleunigt werden. Gegenüber reinen CCM-Architekturen haben Coprozessorlösungen den Nachteil, daß architekturbedingte Flaschenhälse (z.B. der Systembus) erhalten bleiben.

Die (Re-) Konfigurationszeiten von einigen Milli- oder Nanosekunden (je nach Chipgröße, Familie und Konfigurationsmethode) ermöglichen die dynamische Konfiguration des Prozessors. Die vorhandenen Ressourcen können im Sinne einer „virtuellen Hardware“ mehrfach genutzt werden. Der Coprozessor wird so besonders effektiv genutzt.

Bei der Bildverarbeitung haben die zu verarbeitenden Daten ein wesentlich größeres Volumen als der Programm- bzw. Konfigurationscode für deren Verarbeitung. Die Übertragung von relativ kurzen Konfigurationssequenzen verursacht einen vergleichsweise geringeren Kommunkationsaufwand. Es ist also günstiger, die entsprechende Funktionalität zu den Daten zu übertragen und dort die Operationen auszuführen. So könnte ein direkt im Kamera-Interface implementierter FPGA die Daten aufbereiten, bevor sie überhaupt im System verteilt werden.

Coprozessoren müssen an das jeweils umgebende System angepaßt werden. Die Funktionalität (Funktionskern) muß wesentlich häufiger rekonfiguriert werden als die Interfaces zum umgebenden System (Adaptionskomponente). Der Konfigurationsaufwand wird verringert und die Verfügbarkeit steigt, wenn Adaptionskomponente und Funktionskern separat konfigurierbar sind. Dies ermöglichen die Verwendung von partiell konfigurierbaren FPGA oder mehreren einzelnen FPGA.

Für viele Verarbeitungsaufgaben sind mehrere kleine Speicher mit schnellem Zugriff erforderlich (Register, FIFO, Look-up-Tables). Die Belastung des Speicherinterfaces wird verringert, wenn diese Ressourcen direkt auf dem Chip implementiert sind.

3.2 Das DCP-Testbed

3.2.1 Das Hostsystem PADI

Zur Evaluierung des DCP-Konzeptes wurde ein arbeitsfähiges Testbed gebaut. Als Testumgebung dient das Multiprozessorsystem PADI (*P*arallel *A*rchitekture with *Di*stributed Memory), eine Gemeinschaftsarbeit mit der INF FG E der Uni Erlangen [4].

PADI ist entsprechend den Erfordernissen der Echtzeit-Farbbildverarbeitung konzipiert. Das wichtigste Merkmal ist das skalierbare Kommunikationswerk (K-Werk) mit dem systemweiten Kommunikationsbus und lokalen (verteilten) Kommunikationsspeichern für die Prozessorboards. Das K-Werk erreicht eine Datenrate von 108 Mbyte/s. Als Prozessoren werden zur Zeit Transputer verwendet. Das PADI-Konzept ist jedoch offen für den Einsatz beliebiger Prozessoren mit leistungsfähigem Speicherinterface (z.B. PowerPC).

3.2.2 Das DCP-Board

Beim Entwurf des DCP-Boardes wurde besonderer Wert darauf gelegt, daß verschiedene Konzepte für die Nutzung konfigurierbarer Hardware in der Praxis erprobt werden können. Es wurden acht FPGA (Xilinx XC40xx [5]) verwendet. Die Router-Unit aus vier kleineren FPGA (XC4005H) übernimmt die Adaption an das umgebende System und das Management der Da-

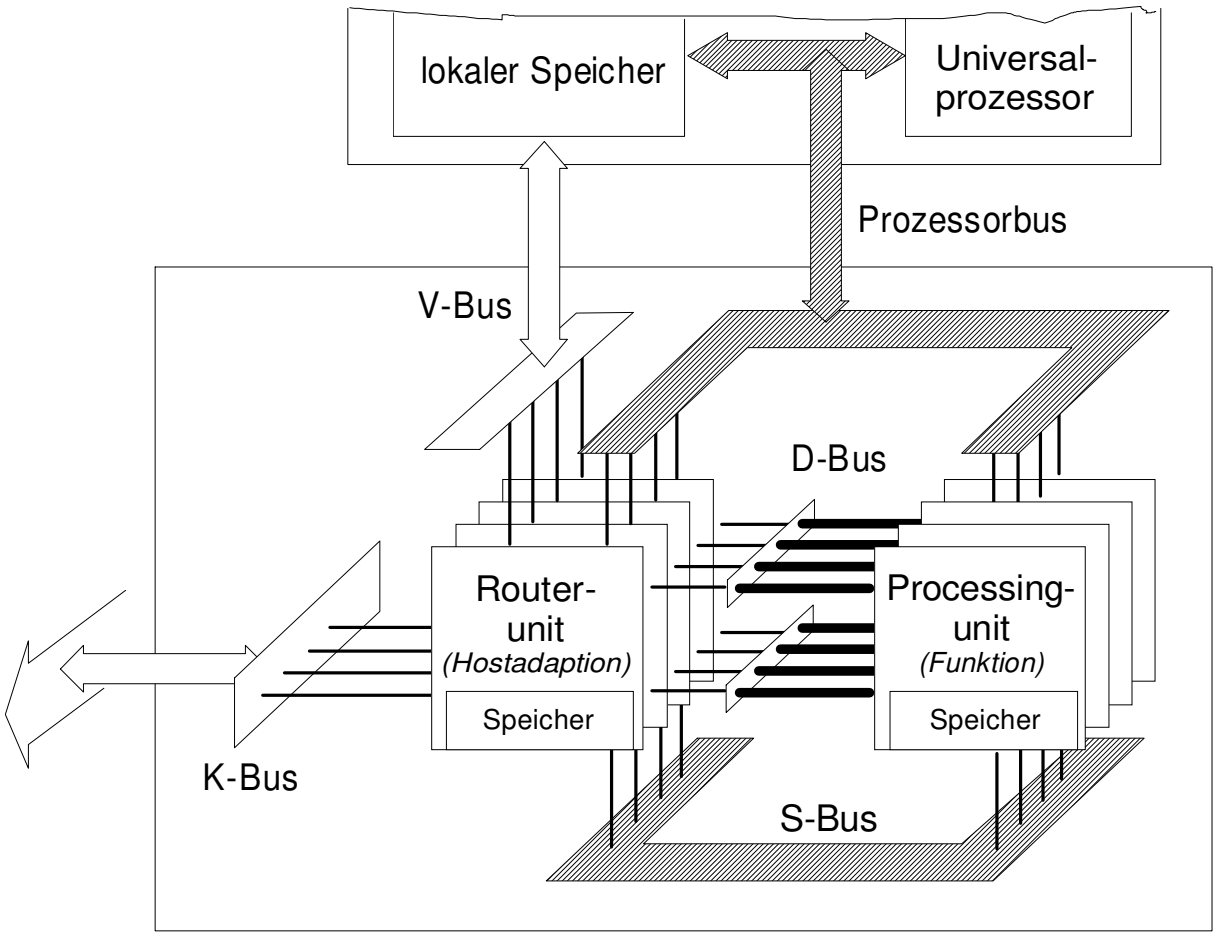

Abbildung 1: Das DCP-Testbed

tenströme. Es bestehen direkte Datenpfade zum systemweiten K-Bus und zum Arbeitsspeicher des lokalen Universalprozessors (V-Bus). Die Verwendung von Dual-Ported-RAM gestattet den gleichzeitigen und unabhängigen Zugriff von DCP und Universalprozessor. Der DCP kann deshalb als shared oder lokaler Coprozessor betrieben werden. Die FPGA's beider Units sind durch einen Datenbus (D-Bus) und den zugehörigen Steuerbus (S-Bus) miteinander verbunden. Die Units und Busse sind in vier getrennt arbeitende Kanäle partitionierbar. In diesem Modus können zum Beispiel die Farbinformationen eines Bildpunktes getrennt verarbeitet werden. Die Ressourcen der Processingunit (vier FPGA XC4013) stehen vollständig für für Verarbeitungsaufgaben zur Verfügung. Die PU können über den D-Bus zu einer langen Pipeline verknüpft werden. Alle FPGA haben lokale Speicher mit schnellem SRAM. Die Konfigurationszeit beträgt für die verwendeten FPGA ca. 30 ms.

Der DCP verfügt über einen programmierbaren Taktgenerator. Der erzeugte Systemtakt kann deshalb flexibel an die jeweilige Konfiguration angepaßt werden.

4 Ergebnisse

Das vorgestellte DCP-Testbed wurde vollständig in Betrieb genommen. Die grundlegende Systemsoftware für Konfiguration und Betrieb des DCP sowie einfache Testanwendungen sind vorhanden. Alle Konfigurationsteile für die Systemfunktionen des Boardes stehen als getestete Module (VHDL-Entitys) zur Verfügung.

Entsprechend dem Einsatz des PADI-Systems wurden bisher vorrangig Algorithmen der Bildverarbeitung portiert. Die Ergebnisse für eine Kantendetektion und einen Objekt-Labeling-Algorithmus sollen kurz vorgestellt werden.

Die implementierte Kantendetektion verwendet den Laplace-Operator. Die eingehenden Bilddaten werden im Datenstrom in ein Binärbild gewandelt und auf Objektkanten untersucht. Der DCP erreichte mit einem PU-Chip, dessen Ressourcen nur zu 25% ausgelastet waren, die notwendige Datenrate für Echtzeitverarbeitung. Der DCP erreicht die gleiche Leistung wie sieben Transputer (T800) oder der Bildverarbeitungsbaustein A110. Letzterer kann jedoch nur einige wenige, fest implementierte Operationen ausführen.

Der Objekt-Labeling-Algorithums ordnet in Echtzeit die einzelnen Punkte Objekten zu. Der verwendete Algorithmus nach Zamperoni [6] ist einfach, seine Implementierung in einen FPGA wird aber durch die erforderliche hohe Datenrate erschwert. Das Verfahren arbeitet zweistufig. In der ersten Phase wird eine Liste von Teilobjekten generiert. In der zweiten Phase wird die Zuordnung der Bildpunkte zu den Objekten bestimmt. Bei dieser Anwendung wird eine PU fast vollständig genutzt. Die Konfiguration arbeitet mit 27 Mhz, die Echtzeitbearbeitung von SVGA-Bildern ist also möglich.

In einer weiteren Arbeit wird das Konzept einer multifunktionalen Prozessorzelle [7] auf dem DCP implementiert. Diese Zelle führt verschiedene Grundoperationen auf der Basis von Bitalgorithmen sehr effizient durch. Einerseits eignen sich dafür die bitorientierten Architekturen der FPGA sehr gut, andererseits wurden Gleitpunktformate bisher auf FPGA-Systemen kaum unterstützt.

Literatur:

[1] Amdahl GM: Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. AFIPS Conference Proceedings 30: 483 - 485, 1967.

[2] Dahm M et. al.: DIBA - Ein interdisziplinärer Ansatz zur Gestaltung des Digitalen Bildarbeitsplatzes für die Röntgenbefundung. Digitale Radiographie - Referate und Vorträge (5. Frankfurter Gespräche über Digitale Radiographie), Schnetztor-Verlag Konstanz: S. 191, 1992.

[3] Spivey M, Page I: How to design hardware with Handel. Technical Report, Oxford University Hardware Compilation Group, 1993.

[4] Oehlrich CW: Ein Multiprozessorsystem zur parallelen Bildverarbeitung auf Transputer-Basis. Erlangen: Universität Erlangen, 1993. - Diss.

[5] Xilinx - The Programmable Logic Company: The Programmable Logic Data Book, 1994

[6] Zamparoni P: Methoden der digitalen Bildsignalverarbeitung. Vieweg Verlag, Braunschweig / Wiesbaden: S. 139, 1989.

[7] Kasche B: Untersuchung geeigneter Algorithmen für Smart-Pixel Rechenwerke. Jena: Universität Jena, 1995. - Diplomarbeit