

| | |
|---------------------|---|
| Zeitschrift: | Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses |
| Herausgeber: | Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen |
| Band: | 80 (1989) |
| Heft: | 1 |
| Artikel: | Geschwindigkeitssteigerung durch Spezialprozessoren am Beispiel der Signalverarbeitung |
| Autor: | Guggenbühl, W. / Mathis, S. |
| DOI: | https://doi.org/10.5169/seals-903620 |

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 12.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Geschwindigkeitssteigerung durch Spezialprozessoren am Beispiel der Signalverarbeitung

W. Guggenbühl und S. Mathis

In diesem Beitrag wird am Beispiel von einigen typischen Algorithmen gezeigt, wie wichtig für die elektronische Signalverarbeitung – wegen ihrer hohen Ansprüche an die Geschwindigkeit – geeignete Prozessorarchitekturen sind. Es wird die Entwicklung vom klassischen Universalprozessor zum Cisc- und Risc-Rechner diskutiert, Beispiele spezieller Architekturen, die Echtzeitanforderungen erfüllen, werden erläutert und die erreichten Rechenleistungen anhand eines typischen Bildverarbeitungsproblems demonstriert.

L'article montre à l'aide de quelques algorithmes typiques l'importance d'une architecture appropriée des processeurs pour le traitement électronique des signaux, à cause de leurs exigences élevées à la vitesse. On discute le développement du processeur universel classique au calculateur Cisc et Risc. Des exemples d'architecture spéciale sont décrits, qui satisfont aux exigences d'un système en temps réel, et les performances ainsi obtenues mises en évidence à l'aide d'un problème typique de traitement des images.

Adresse der Autoren

Prof. Dr. Walter Guggenbühl und dipl. El.-Ing. Severin Mathis, Institut für Elektronik, ETH-Zentrum, 8092 Zürich.

Die Mechatronik stellt zunehmend höhere Anforderungen an die Verarbeitungsgeschwindigkeit der elektronischen Systemteile. So verlangt z.B. die Steuerung und Regelung bewegter Maschinenteile immer rechenintensivere Verfahren, und auch die Sensoren sowie die nachgeschalteten Verarbeitungsstufen erfordern immer aufwendigere Prozeduren. Ein besonders interessantes Anwendungsgebiet dieser Art ist die Robotertechnik, in der optische und taktile Sensoren sowie moderne Steuer- und Regelungsverfahren eine zentrale Rolle spielen.

Der rasche technologische Wandel der Elektronik, insbesondere der Übergang von der Analog- zur Digitaltechnik und damit der zunehmende Einsatz von ins System eingebetteten Computern, prägt natürlich auch dieses Anwendungsgebiet. Nun stossen aber die Anforderungen der Elektroniker bei den in Echtzeit zu lösenden Signalverarbeitungsaufgaben, wie sie in der Mechatronik auftreten, sehr oft an die Grenze der Arbeitsgeschwindigkeit klassischer Computertechnik. Dies gilt insbesondere für die seit Mitte der siebziger Jahre eingesetzten Mikroprozessorchips, deren Architektur auf den Konzepten der Minirechner der sechziger Jahre beruht. Bekanntlich wird die universelle Verwendbarkeit dieser Schaltungen mit Verarbeitungszeit bezahlt; ausserdem führt die Programmierung mit Hochsprachen zu weiteren Geschwindigkeitsverlusten, die je nach Compilergüte grosse Werte annehmen können.

Anspruchsvolle Signalverarbeitungsaufgaben wie z.B. die Bildverarbeitung sind deshalb mit üblichen Mikroprozessoren nicht zu lösen; sie erfordern spezielle Verarbeitungsschaltungen, deren Palette je nach Geschwindigkeitsanforderungen von so genannten Signalprozessoren bis zu Lösungen, die spezifisch auf einen ein-

zigen Algorithmus zugeschnitten sind, reicht. Diese wichtige Frage des Ausmasses der notwendigen Spezialisierung von Signalverarbeitungsarchitekturen ist Thema dieser Arbeit. Am Institut für Elektronik der ETH Zürich sind in Fortsetzung des in den siebziger Jahren erfolgten Aufbaus der Mikroprozessortechnik seit einiger Zeit zahlreiche Untersuchungen zu diesem Geschwindigkeitsproblem durchgeführt und Implementationsbeispiele realisiert worden. Im Rahmen dieser Aufgaben haben moderne Signalprozessoren einen besonderen Stellenwert. Zahlreiche dieser Arbeiten fanden im Gebiet der Mechatronik Verwendung, wie das Bildaufnahmesystem für einen Pingpong-Roboter [1], ein Zustandsregler für den Antrieb eines hydraulischen Kolbens [2] und eine auf dem Markt erhältliche Bildverarbeitungskarte [3].

Anforderungen an die Rechenleistung am Beispiel typischer Signalverarbeitungsalgorithmen

Die wohl häufigste Signalverarbeitungsfunktion ist die Filterung. Sie wird im Falle digitaler Signale mittels algorithmischer Verarbeitung zeitdiskreter Abtastwerte durchgeführt, d.h. es wird die Differenzengleichung $y[nT] = f(x[nT])$ gelöst (Fig. 1). Im Fall der sogenannten Transversalfilter lautet diese Gleichung

$$y[nT] = \sum_{k=0}^{K-1} b_k \cdot x[(n-k)T] \quad (1)$$

mit $T = \text{Abtastzeit}$

Sie entspricht der zeitdiskreten Faltung des Signals $x[nT]$ mit der Impuls-

antwort $b[kT]$ des betreffenden Filters, wobei T die Abtastzeit bezeichnet. Im Falle der rekursiven Filter ist die allgemeinere Differenzengleichung

$$y[nT] + \sum_{m=1}^M a_m \cdot y[(n-m)T] = \sum_{k=0}^K b_k \cdot x[(n-k)T] \quad (2)$$

zu lösen.

Auch für «zweidimensionale» Signale (z.B. Bildsignale) existieren analoge Problemstellungen. Die transversale Filteroperation wird hier als Lokaloperation, das der Impulsantwort $b[kT]$ entsprechende Muster als Maske $b[k,i]$ bezeichnet (Fig. 2).

Der Bildpunkt mit den Koordinaten $[m,n]$ im transformierten Bild P' errechnet sich aus den Punkten des Originalbildes P gemäss

$$P'[m,n] = A_0 + B_0 \cdot \sum_{k=-s}^s \sum_{i=-t}^t b[k,i] \cdot P[m+k, n+i] \quad (3)$$

Die Lokaloperation wird bei der Vorverarbeitung von Bildern, z.B. zur Filterung, Rauschunterdrückung, Kantendetektion und Korrelation, verwendet.

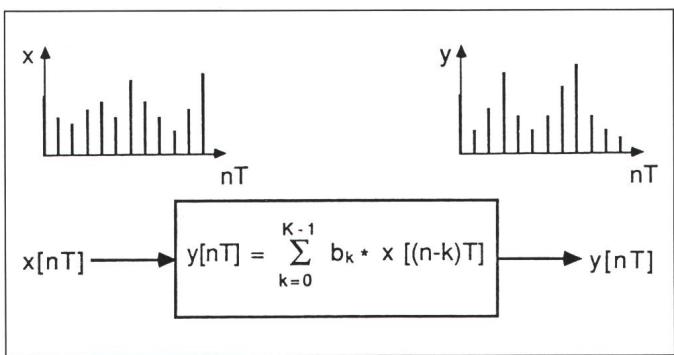
Die oben beschriebenen Filteroperationen benötigen nach Gl. (1), (2), (3) lediglich die Operationen Multiplikation und Addition bzw. Multiplikation und Akkumulation. Die nachfolgenden Zahlen geben einen Eindruck der benötigten Rechenleistung.

Beispiel 1: Ein Transversalfilter der Ordnung $K = 10$ für die Sprachverarbeitung benötigt bei einer Abtastrate von 8 kHz je $8 \cdot 10^4$ Multiplikationen, Additionen und Datentransporte pro Sekunde.

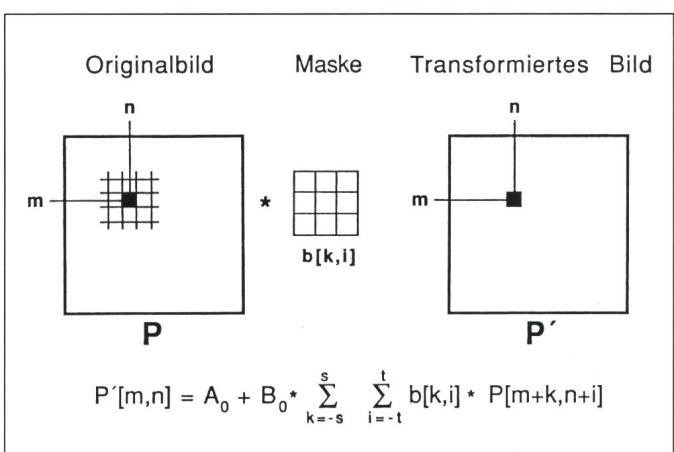
Beispiel 2: Die Verarbeitung eines Bildes mit 256×256 Punkten mit einer Maske 3×3 benötigt 650 000 Multiplikationen und ebensoviele Additionen und Datentransporte. Bei der Echtzeit-Vorverarbeitung einer Bildsequenz mit 50 Halbbildern pro Sekunde müssten demzufolge 32,5 Millionen Additionen, Multiplikationen und Datentransporte pro Sekunde ausgeführt werden.

Bei der Implementation der entsprechenden Algorithmen in einem Com-

Figur 1
Faltung eines zeitdiskreten Signals
 b_k ist die zeitdiskrete Impulsantwort des Netzwerks



Figur 2
Zweidimensionale Filterung eines gerasterten Bildes; Lokaloperation mit der Maske $b[k,i]$



puter kommen dazu noch Operations schritte für die Schleifensteuerung und Adressrechnung. Die genannten Zah len, insbesondere diejenigen für die Bildverarbeitung, liegen jenseits der Leistungsfähigkeit heutiger universeller Mikroprozessoren.

Architektur des universellen Mikrorechners

Die Entwicklung der Mikroprozessoren erfolgte im wesentlichen nach dem Vorbild der Minirechner der sechziger Jahre. Das in Figur 3a vereinfacht dargestellte Datenverarbeitungswerk des klassischen Rechners basiert auf der programmierbaren Verknüpfung zweier Datenströme in der ALU (Arithmetic Logic Unit). Die Daten werden aus dem Speicher ausgelesen und die Resultate dort wieder abgelegt.

In den klassischen ALU können in einem Verarbeitungsschritt (Taktzyklus) nur relativ einfache Datenmanipulationen ausgeführt werden, im Falle der arithmetischen Operationen sind dies Addition, Subtraktion, Zahleninversion und Zahlenvergleich. Die für Signalverarbeitung notwendige

Multiplikation muss in solchen einfachen Systemen aus einer Sequenz von Additionen und Zahlenverschiebungen zusammengesetzt werden und ist entsprechend zeitaufwendig.

Neben der in Figur 3a vereinfacht dargestellten Verarbeitungseinheit ist das Steuerwerk der zweite wesentliche Bestandteil eines Prozessors. Es erzeugt die dem Maschinenprogramm entsprechenden Steuerwörter für das Verarbeitungswerk, welche die Operationsart sowie die Adressen der beteiligten Operanden festlegen (Fig. 3b). Ein Teil dieser Befehlswörter dient außerdem steuerwerksintern dem Programmablauf, der vom jeweiligen Datenzustand abhängig gemacht werden kann (z.B. Verzweigungen auf Grund des Status).

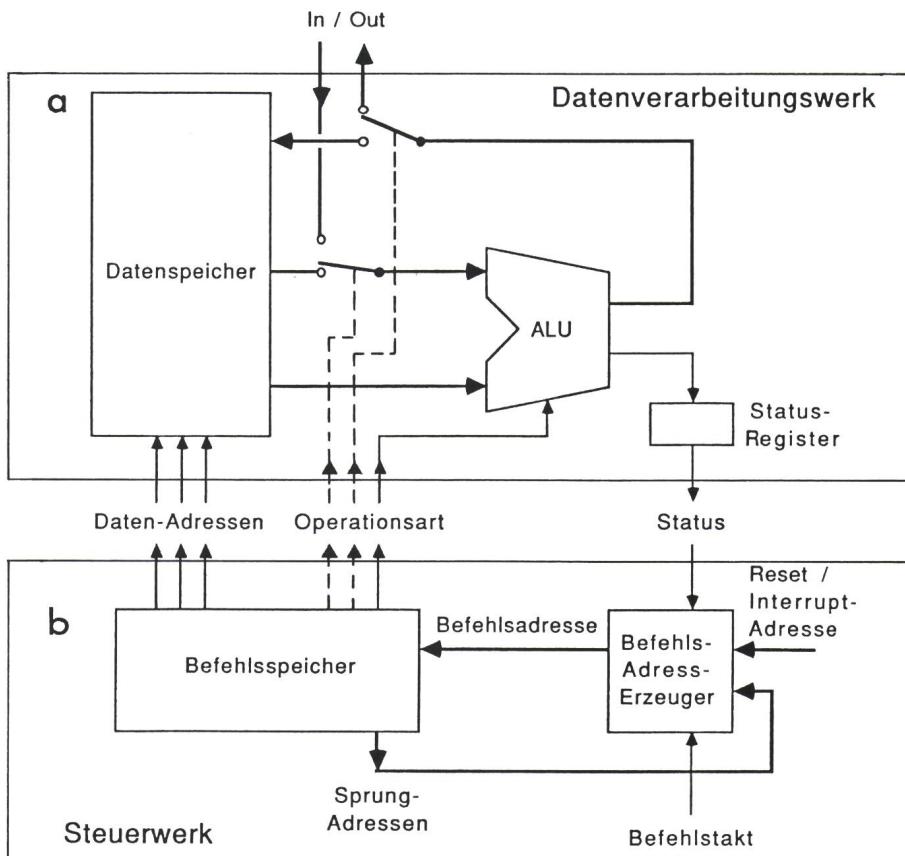
Die steigende Integrationsdichte ermöglichte in den frühen achtziger Jahren den Vorstoß der Mikroprozessoren in den Leistungsbereich der Midi und Grossrechner vorangehender Generationen, wobei die Konzepte dieser Vorbilder im wesentlichen übernommen wurden, wie z.B. die folgenden typischen Architekturmerkmale:

Daten- und Befehlsspeicher sind zusammengelegt (von-Neumann-Archi-

tekur) und hierarchisch in Register, Cache, Arbeitsspeicher und Hintergrundspeicher aufgeteilt.

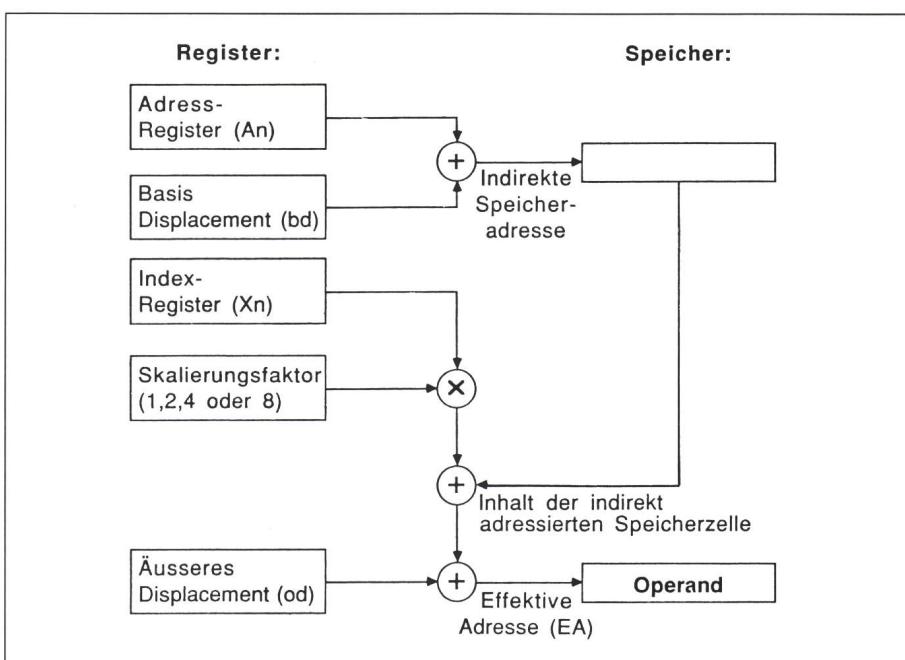
Die Kluft zwischen den Hochsprachen und der Hardwarearchitektur (sog. Semantic Gap) wurde durch verschiedenste Konzepterweiterungen verkleinert; zu diesen Massnahmen gehören beispielsweise die Unterstützung des Adresshaushalts bei Prozeduraufrufen durch das Stapelspeicherprinzip (Stack). Der Befehlssatz wurde durch immer komplexere Instruktionen erweitert, mit dem Ziel, Objektprogramme mittels möglichst wenigen Befehlswörtern im Speicher darstellen zu können und damit Platz und Abholzeit der Befehle aus dem relativ langsamem Arbeitsspeicher zu sparen. Ein wesentlicher Schritt in diese Richtung ist die Mikroprogrammierung, d.h. die Auslösung grösserer Sequenzen elementarer Operationsschritte im Prozessor durch einen Maschinenbefehl. Sie ermöglicht neben der effizienten Programmierung der erwähnten komplexen Datenoperationen auch die Durchführung umfangreicher, den Datenstrukturen moderner Hochsprachen angepasster Adressrechnungen. Die Figur 4 illustriert ein solches mehrstufiges Adressmodifikationsschema am Beispiel des Mikroprozessors MC 68020. Durch Erweiterungen dieser Art entstanden Computersysteme (auch Mikrocomputer) mit sehr umfangreichem Befehlssatz, sogenannte Complex Instruction Set Computer (Cisc).

Die oben beschriebene Entwicklungsrichtung im Computerbau wurde in den letzten Jahren zunehmend in Frage gestellt, vor allem dort, wo es um zeitkritische Anwendungen geht. Entwicklungstrends in der Halbleiter- und Softwaretechnik (Speicherpreise, VLSI-Technologie, optimierende Compiler) regten viele Forscher zu einer Neuüberprüfung der den Architekturentwicklungen zugrunde liegenden Optimierungsziele an. Die Analyse von Befehlsfolgen compilierter Pascal- und C-Programme zeigte, dass meistens eine kleine Anzahl einfacher Befehle sehr häufig benutzt wird, die jedoch gemessen an ihrem Verarbeitungsbeitrag grosse Ausführungszeiten benötigen, da das Steuerwerk für selten gebrauchte komplexe Operationen ausgelegt ist. Mit dem Begriff Risc (Reduced Instruction Set Computer) wurde die Idee «zurück zur einfachen Architektur» propagiert, die in letzter Zeit von zahlreichen Herstellern aufgenommen wurde. Die Risc-Anhänger



Figur 3 Datenverarbeitungswerk und Steuerwerk

- a Struktur eines einfachen klassischen Datenverarbeitungswerks: Das Ergebnis einer in der ALU ausgeführten Operation kann das Verhalten des Steuerwerks beeinflussen (Status).
- b Vereinfacht gezeichnete Struktur eines Steuerwerks: Es erzeugt im wesentlichen die Steuerbefehle für das Datenwerk. Sie legen das «Wo» (Datenadressen) und das «Wie» (ALU-Funktion sowie Datenpfadschaltung = Operationsart) der Datenverarbeitung fest. Einige Befehle haben lediglich steuerwerksinterne Auswirkungen, z.B. Sprungbefehle aufgrund des Datenwerk-Status.



Figur 4 Beispiel einer mehrstufigen Adressrechnung beim Prozessor MC 68020

Die Adresse EA des Operanden im Speicher berechnet sich nach der Formel $EA = [bd + [AN]] + [Xn] \cdot scale + od$, wobei die eckige Klammer den Inhalt des entsprechenden Registers oder Speichers bezeichnet. Dieses Prozedere lässt sich z.B. zum Adressieren eines einzelnen Array-Elements in einer Liste einsetzen. Jedes Element dieser Liste kann dabei ein Datenrekord, bestehend aus mehreren Arrays, sein.

sehen die optimale Architektur etwa wie folgt:

- wenige einfache Instruktionen von möglichst einheitlicher Länge
- Ausführung in einem bis zwei Taktzyklen,
- wenige Adressierungsarten,
- Register-Register-Architektur, d.h. lediglich Transportbefehle zum und vom Speicher und Rechenoperationen nur zwischen Registern in der Verarbeitungseinheit,
- grosse Registerzahl,
- schnelles Steuerwerk ohne Mikroprogrammierung,
- Instruktions-Cache-Speicher zur Verminderung der Arbeitsspeicherzugriffe beim Abarbeiten von Schlaufen.

Ein grosser Teil des Semantic Gap zwischen Hochsprache und Maschine wird dadurch überbrückt, dass anstelle von sehr leistungsfähigen, optimierenden Compilern für die Bearbeitung von komplexen Operationen und des zeitaufwendigen Transports von Befehlen und Daten zwischen Arbeitsspeicher und Prozessor ein Befehls cache und ein grosser Daten-Registersatz auf dem Prozessorchip integriert ist. Die Tabelle I vergleicht die typischen Daten eines Cisc- und eines Risc-Rechners. Da Signalverarbeitungsalgorithmen dank ihrer regulären, einfachen Struktur von den durch allgemeine Datenverarbeitungsaufgaben geprägten Eigenschaften der Cisc-Rechner wenig Gebrauch machen, scheint ihre Implementation auf Risc-Rechnern effizienter zu sein.

Neben den skizzierten Extremallo-lungen Cisc und Risc sind zahlreiche Mischformen realisiert worden. Ein besonders interessanter Ansatz ist der sogenannte *Transputer*, in dem zahlreiche Risc-Eigenschaften (kurze Befehslänge, wenige Adressiermodi) unter Beibehaltung einiger Cisc-Merkmale, z.B. der klassischen Mikropro-

grammierung, implementiert sind. Ein besonders nützliches Subsystem des Transputers ist die parallel zur Datenverarbeitung laufende Kommunikationseinheit, die den Informationsaustausch mit Nachbarprozessoren steuert. Der Transputer ist damit ein interessanter Lösungsansatz für Multiprozessorsysteme, die angesichts der stetig wachsenden Anforderungen an die Verarbeitungsleistung immer mehr Bedeutung gewinnen.

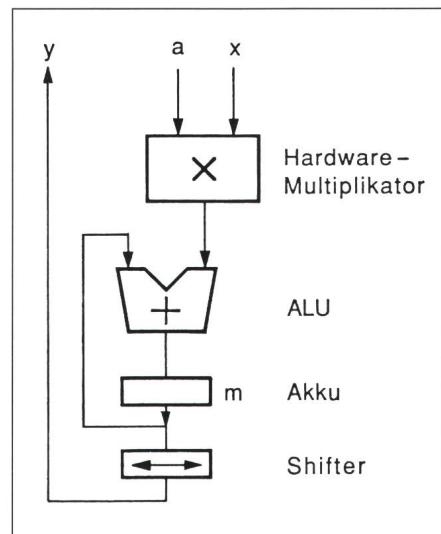
Spezialarchitekturen für die Signalverarbeitung

Die den Risc-ähnlichen Architekturen zugrunde liegenden Ideen schliessen die Verwendung spezieller, spezifische Funktionen unterstützender Hardware im Prozessorkonzept sowohl auf der Verarbeitungs- wie auf der Steuerwerkseite nicht aus. Es ist naheliegend, dass solche Massnahmen die Verarbeitungsleistung wesentlich steigern können.

Digitale Signalprozessoren (DSP)

Die unter diesem Namen seit einiger Zeit angebotenen Spezialarchitekturen sind für die Bearbeitung digitaler Signale optimiert. Die klassische Computerarchitektur wurde beim Signalprozessor typischerweise in folgenden Punkten an die auf diesem Gebiet üblichen Algorithmen angepasst:

- Erweiterung des Verarbeitungswerks um einen schnellen Hardware-Multiplikator und eine Schiebeeinheit, die die Abarbeitung der MAC-Operation (Multiply and Accumulate) in einem TaktSchritt erlauben (Fig. 5),
- grosse On-Chip-Speicher für Daten- und Filterkoeffizienten,
- spezialisierter Adressbildungsrechner, d.h. ein Rechner, der aufgrund



Figur 5 Vereinfachte Datenwerkstruktur eines Signalprozessors, optimiert für die MAC-Elementaroperation ($m \leftarrow a \cdot x + m$)

Diese Struktur erlaubt die Mehrfachausführung dieser Operation, wie dies bei der Aufsummierung von Produkten notwendig ist (z.B. Fig. 1).

der Kenntnis von speziellen Algorithmenstrukturen (z.B. Fast-Fourier-Transformation, Ringpuffer) die nötigen Daten (z.B. Filter- und Signaldaten) besonders effizient aus dem Speicher ausliest,

- getrennte, zeitlich parallel betriebene Datenverarbeitungs- und Adressrechner,
- Verwendung von getrennten Befehls- und Datenspeichern (sog. Harvard-Architektur) zur Fliessbandverarbeitung (Pipelining) von Befehlen und Daten.

Die Tabellen II und III zeigen, dass aus einer solchen Spezialisierung der Prozessoren typische Geschwindigkeitsgewinne von mehr als einer Größenordnung resultieren. Es sei jedoch darauf hingewiesen, dass diese Vorteile verschwinden, wenn man die betreffenden Prozessoren außerhalb der Anwendungsbereiche, für die sie optimiert sind, einsetzt (z.B. für Rechnungen mit doppelter numerischer Präzision).

Fliesskommarechnung durch Hardware

Die noch immer steigende Komplexität der auf einem Chip integrierbaren Schaltungen erlaubt in neuester Zeit, das MAC-Rechenwerk von Signalprozessoren ohne wesentliche Durchsatzeinbusse auf das 32-Bit-Fliesskommaformat (Floating-Point) zu erweitern. Damit wird der Program-

| Merkmal | MC 68030 | Am 29000 |
|------------------------|--------------|----------|
| Anzahl Instruktionen | 174 | 115 |
| Instruktionswortlänge | 16...176 Bit | 32 Bit |
| Instruktionsformat | variabel | fix |
| Register | 16 | 192 |
| davon: frei verwendbar | | 192 |
| Datenregister | 8 | |
| Adressregister | 8 | |
| Adressierungsarten | 18 | 4 |

Tabelle I Vergleich zwischen einem CISC- und einem RISC-Mikroprozessor

mierer von der schwierigen Skalierarbeit und allfälliger Verwendung der oben erwähnten Methode zur Präzisionserweiterung entbunden. Mit Fliesskommaverarbeitung eröffnen sich den Signalprozessoren neue Anwendungsgebiete.

Hardwareunterstützte Fliesskommaverarbeitung – durch Coprozessoren – ist im Bereich der Universalprozessoren bereits seit längerer Zeit verbreitet. Ob entsprechende Blöcke auf dem Hauptprozessorchip integriert werden sollen, um einen Geschwindigkeitsgewinn zu erreichen, ist eine Frage nach der Priorität, d.h. nach der Wichtigkeit der Funktionen, die auf dem Zentralchip eines Systems unterzubringen sind.

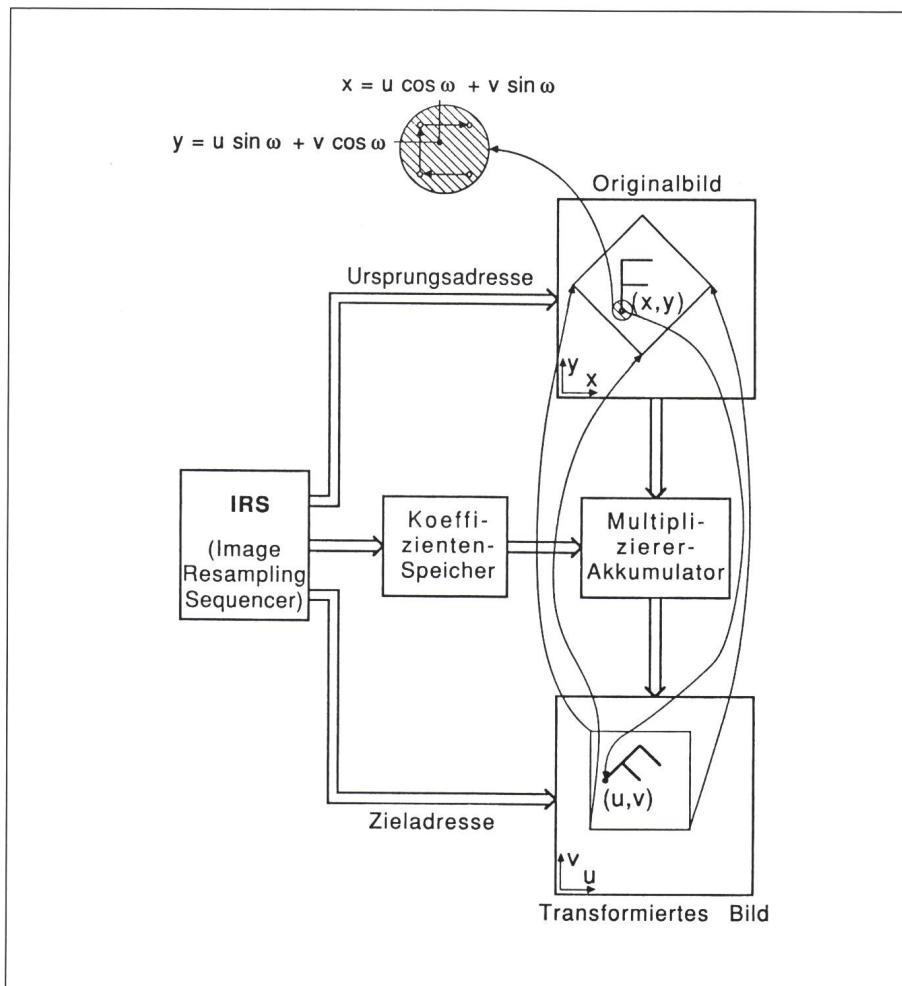
Spezielle Adressierwerke

Zur hohen Verarbeitungsleistung trägt nicht nur das spezialisierte Datenverarbeitungswerk, sondern ebenso eine entsprechend angepasste parallel laufende Adressaufbereitungseinheit bei.

Das in einem optimierten Datenadressrechenwerk steckende Beschleunigungspotential sei am Beispiel des CMOS Image Resampling Sequencers TMC 2301 illustriert (Fig. 6). Diese Einheit ist für die Transformation von gerasterten Bildern optimiert (z.B. Rotation, Vergrösserung, Kompression). Die betreffende Operation erfolgt zwischen zwei Bildspeichern. Ausgehend von den Koordinaten eines im Zielspeicher darzustellenden Bildpunktes wird die Adresse des Urbildpunktes im Quellspeicher errechnet. Da wegen der Rasterung bei den meisten Abbildungsarten (z.B. Bildrotation) keine punktweise Zuordnung zwischen ursprünglichem und transformiertem Bild besteht, wird die Helligkeit des Zielpunktes durch eine optimierte Interpolation der Helligkeitswerte eines Bildfeldes wählbarer Grösse errechnet, wobei dieses spiralförmig aus dem Quellspeicher ausgelesen wird. Dabei werden völlig andere Adressierschemata als bei den in Hochsprachen üblichen Datenstrukturen verwendet. Mit dem oben genannten Sequenzer, der dieser speziellen Aufgabe angepasst ist, lassen sich Rotationen eines Bildes von 512×512 Punkten in 63,5 ms durchführen.

Spezielle Verarbeitungseinheiten

Ein Signalprozessor benötigt zur Berechnung eines Abtastwertes am



Figur 6 Bildrotation mit dem Image Resampling Sequencer (IRS)

Für jeden Bildpunkt (u,v) des transformierten Bildbereichs werden im IRS die entsprechenden Koordinaten (x,y) im Originalbild berechnet. Da diese nicht ins Punktraster des Originalbildes fallen, werden durch einen spiralförmigen «Walk» die Nachbarpunkte des Originalbildes berechnet. Die Intensität des transformierten Pixels wird dann durch Interpolation aus den betreffenden Intensitätswerten berechnet (z.B. aus vier Nachbarn). Für jedes transformierte Pixel müssen somit mehrere Originalpixel ausgelesen werden. Die Generierung der benötigten Adressen wird dabei durch den IRS automatisch durchgeführt.

| Prozessor | Taktfrequenz [MHz] | Ausführungszeit [ns] |
|------------|--------------------|----------------------|
| MC 68000 | 12 | 6500 |
| MC 68020 | 16 | 1940 |
| MC 68030 | 20 | 1500 |
| T212 | 20 | 1200 |
| T414, T800 | 20 | 1700 |
| Am 29000 | 25 | 760 |
| TMS 32010 | 20 | 400 |
| TMS 320C25 | 40 | 200 |
| TMS 320C30 | 33 | 120 |

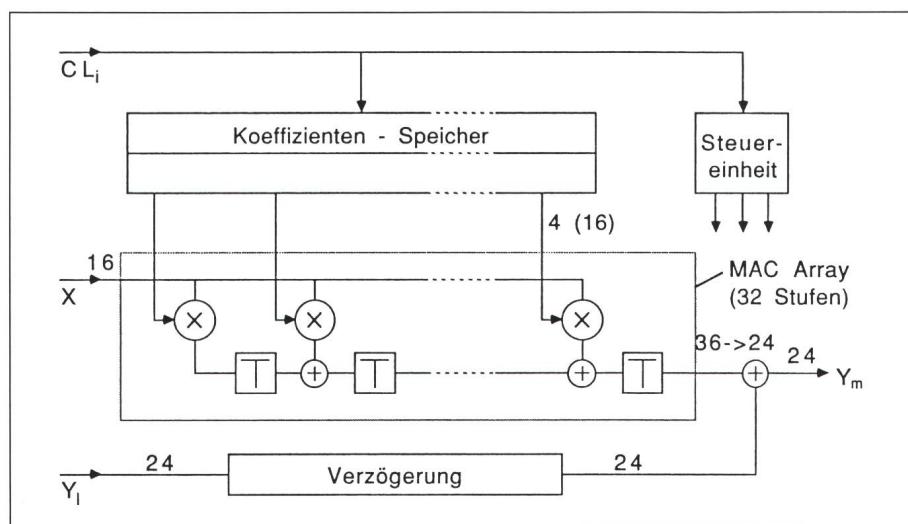
Prozessortypen:
MC 680XX
typischer Cisc-Prozessorchip, evtl. zusätzlich verlangsamt durch Datentransportoperationen
T 212
Transputer mit 16-Bit-Operanden
T 414
Transputer mit 32-Bit-Operanden
T 800
Transputer mit 32-Bit-Operanden
AM 29000
Risc-Architektur
TMS 320XXX
Signalprozessoren
(MAC-Operationssequenzen sind durch Schlaufenbefehle zusätzlich beschleunigbar)

Tabelle II Ausführungszeiten der Multiplikation-Additions-Operation zwischen Registern
(16-Bit-Operanden, 32-Bit-Resultat, Fixkomma)

Filterausgang gemäss Gleichung (1) K Multiplikations-Additions-Schritte, wobei K die Länge des betreffenden Transversalfilters bezeichnet. Grundsätzlich können diese Operationen zeitlich parallel ausgeführt werden, wenn die Multiplikations-Additions-Struktur hardwaremässig vervielfacht und gemäss Figur 7 zusammengeschaltet wird. Mit einer solchen Struktur, die heute in den Konfigurationen von 32-MAC-Einheiten für eindimensionale Signale und 64 in Matrix (z.B. 8×8) verschaltbaren MAC-Einheiten für zweidimensionale (Bild-)Signale auf dem Markt angeboten wird, stösst man punktuell in den Leistungsbe- reich grosser Vektorrechner vor (rund 10^9 Multiplikationen/Additionen pro Sekunde). Diese enorme Rechenleistung auf einem Chip ist jedoch nur auf Kosten der Flexibilität, d.h. durch Spezialisierung auf praktisch einen einzigen Verarbeitungsalgorithmus möglich. Solche algorithmenspezifische Höchstleistungen können in Zukunft auch mit applikationsspezifischen integrierten Schaltungen (ASICS), in deren Bibliotheken Addierer und Multiplizierer als Makroblöcke angeboten werden, erreicht werden.

Flexible Signalverarbeitungsarchitekturen

Am Institut für Elektronik wurde die für Echtzeit-Bildverarbeitung notwendige Rechenleistung auf einem alternativen Weg erbracht, der eine grössere Flexibilität als die algorithmenspezifischen Schaltungen bietet. Die Signalverarbeitung erfolgt über Speichereinheiten, in denen die Verknüpfungsoperationen als Tabellen abgespeichert sind. Diese als verallgemeinerte ALU auffassbaren Blöcke (Fig. 8a) können beliebige (auch nicht-lineare) Kombinationen der beiden als Adressen angelegten Eingänge erzeugen. Kompliziertere Funktionen, wie sie z.B. die 3×3 -Lokaloperation darstellt, werden in einer Fliessband-(Pipeline-)Anordnung solcher ALU-Blöcke (Fig. 8b) abgearbeitet. Die einzelnen Blöcke in diesem Sydama-Rechner können durch ein programmierbares busartiges Netz verbunden werden. Der Rechner wird dadurch zum programmierbaren Bildverarbeitungssystem, mit dem Algorithmen in Echtzeit erprobt werden können [4]. Diese flexibel einsetzbaren Recheneinheiten sind mit algorithmenspezifischen Blöcken kombinierbar.



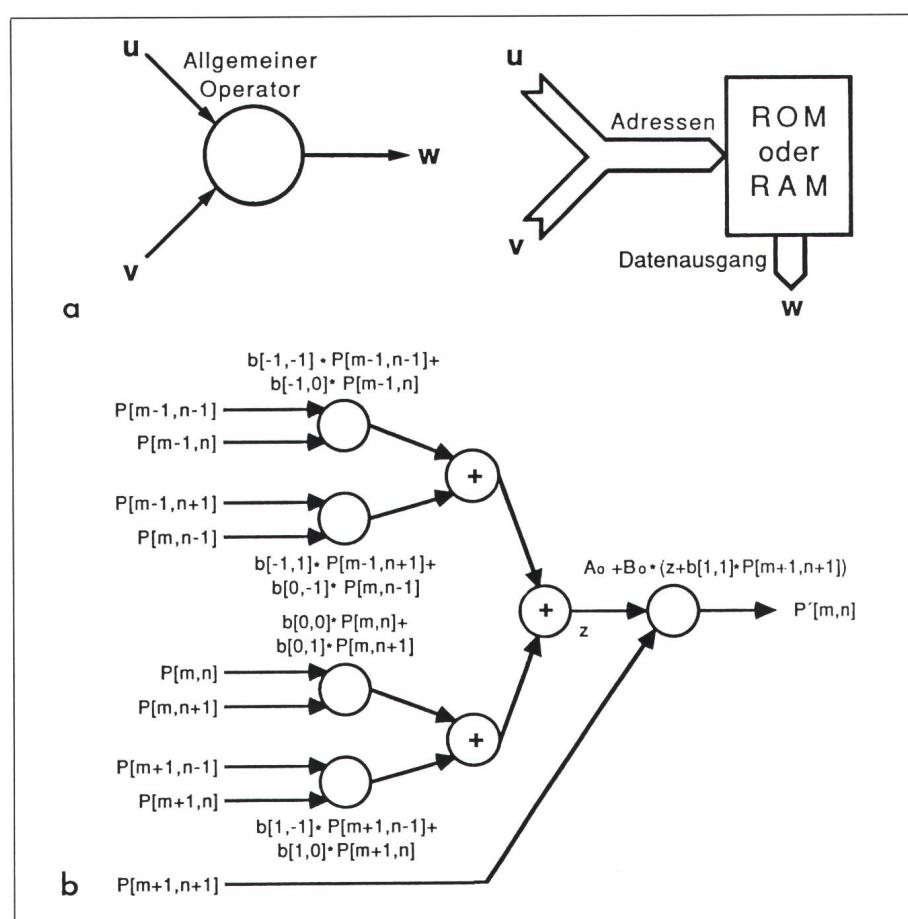
Figur 7 Blockschema einer kommerziell erhältlichen MAC-Array-Schaltung

- × 16×4 -Bit-Multiplikator; 16-Bit-Daten, 4-Bit-Koeffizienten; wählbar sind auch 16-Bit-Koeffizienten, wenn Einbusse an Ausführungsgeschwindigkeit in Kauf genommen wird.

- + 36-Bit-Addition und Reduktion des Ergebnisses auf 24 Bit

- T Verzögerungsglied

Bei 4-Bit-Koeffizienten und 16-Bit-Daten beträgt die Zykluszeit 100 ns. Dies bedeutet, dass alle 100 ns das Ergebnis von 32 Multiplikationen/Additionen ausgegeben wird. Die Schaltung berechnet somit 320 Mio MAC-Operationen pro Sekunde.



Figur 8 Flexible Signalverarbeitungsarchitektur

- a Elementarer Operator in der Form einer Look-up-Tabelle im Speicher (Operanden = Halbadressen, Verknüpfungsresultat = Datenausgang). Er ermöglicht die Realisierung linearer und nicht-linearer Operationen mit zwei Eingangsvariablen, z.B. $w = a \cdot u + b \cdot v$ (a, b = Konstante).

- b 3×3 -Lokaloperation (Fig. 2), aufgebaut mit Operatoren aus 9a. Die $b(k,i)$ -Koeffizienten beziehen sich auf eine 3×3 Matrix ($-1 < k < +1; -1 < i < +1$).

| Sprache | Rechner | Taktfre-quenz [MHz] | CPU | Zeit [s] |
|--------------|--------------|---------------------|---------------|------------------|
| Turbo-Pascal | IBM-XT | 4,7 | 8088 | 85,0 |
| Turbo-Pascal | Olivetti M24 | 8,0 | 8086 | 36,0 |
| Assembler | Olivetti M24 | 8,0 | 8086 | 12,0 |
| Turbo-Pascal | HP Vectra | 10,0 | 80286 | 12,0 |
| Assembler | | 8,0 | 68000 | 9,0 ¹ |
| Fortran | VAX 11/780 | | | 5,4 ¹ |
| Forth | | 4,7 | NC 4000 | 5,3 |
| Assembler | HP Vectra | 10,0 | 80286 | 3,7 |
| Fortran | VAX 11/785 | | | 3,2 ¹ |
| Forth | | 4,7 | NC 4000 | 2,9 ² |
| Assembler | | 16,0 | 68020 | 2,0 ¹ |
| Occam | Transputer | 20,0 | T414 | 1,0 |
| Assembler | Signalproz. | 40,0 | TMS320C25 | 0,3 |
| IPL | Sydama | 10,0 | Spezialhardw. | 0,02 |

Tabelle III Ausführungszeiten verschiedener Implementationen eines 3×3 -Lokaloperators über ein Bild von 256×256 Punkten.

¹ hochgerechnete Werte

² mit externem Hardware-Multiplikator

Quantitative Illustration der Leistungssteigerung

Die Leistungsfähigkeit der in den vorangegangenen Abschnitten qualitativ diskutierten Architekturvarianten soll im folgenden quantitativ illustriert werden. Die Zahlen beruhen zum grossen Teil auf Messungen an Architekturvarianten, die im Institut für Elektronik der ETH Zürich implementiert wurden. Einige dieser Werte sind auf Grund von Teilversuchen hochgerechnet. Die Tabelle II gibt die für eine MAC-Operation (Multiply Accumulate) in verschiedenen Prozessoren notwendige Ausführungszeit. Dabei ist der Aufwand z.B. für die Einbettung in Schlaufen und für die

Adressrechnung der zugehörigen Daten nicht einbezogen.

Um einen Eindruck über den Einsatz der verschiedenen Konzepte im Rahmen eines Algorithmus zu erhalten, sind in Tabelle III die Ausführungszeiten verschiedener Implementationen des weiter vorne (Fig. 2) beschriebenen 3×3 -Lokaloperators für die Bildvorverarbeitung angegeben. Man erkennt deutlich die Auswirkung des Architekturunterschiedes sowie auch die durch die Compilation eingebrachten Verluste.

Zusammenfassung und Schlussfolgerungen

Die für digitale Signalverarbeitung in Echtzeit benötigte Rechenleistung

kann im Falle der Audiosignale durch Signalprozessoren erbracht werden. Diese heute auf Filteralgorithmen ausgerichteten Prozessorarchitekturen werden bereits weitgehend praktisch eingesetzt. Zurzeit findet auf der Entwicklungsseite der Übergang von den fixkommaverarbeitenden Strukturen auf hardwaremässig implementierte Gleitkomma-Einchip-Prozessoren statt.

Die für die Bildverarbeitung, insbesondere die Vorverarbeitungsalgorithmen notwendigen Rechenleistungen erfordern zur Quasi-Echtzeit-Verarbeitung heute noch zusätzliche, spezialisierte Einheiten. Um eine genügende Leistungssteigerung zu erreichen, müssen sowohl die Datenverarbeitungs- wie auch die Adressrechenseite der Prozessoren optimiert werden. Bildvorverarbeitung im Fernsehtakt ist, wie an Beispielen in diesem Aufsatz gezeigt wurde, bei entsprechendem Aufwand heute grundsätzlich möglich.

Literatur

- [1] H.P. Fässler, K. Buffinton and E. Nielsen: Design of a high-speed robot skilled in the play of ping-pong. 18th International Symposium on Industrial Robots, 26...28 April 1988, Lausanne.
- [2] S. Mathis: Digitaler Zustandsregler mit Signalprozessor. Mechatronik Symposium, 10. und 11. September 1987, ETH Zürich. Vorabdruck der Referate. Zürich, Schweizerische Gesellschaft für Mikrotechnik (SGMT) und Eidgenössische Technische Hochschule (ETH), 1987.
- [3] P. Aschwanden: Real-time tracker with signal processor. European Signal Processing Conference (EUSIPCO) 5...8 September 1988, Grenoble.
- [4] A. Gunzinger, S. Mathis und W. Guggenbühl: Synchroner Datenflussrechner zur Echtzeitbildverarbeitung. Bull. SEV/VSE 79(1988)7, S. 362...367.