

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: 77 (1986)

Heft: 11

Artikel: Ein-Chip-Signalprozessoren in der Mess- und Regelungstechnik

Autor: Hanselmann, H.

DOI: <https://doi.org/10.5169/seals-904213>

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. 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. [Siehe Rechtliche Hinweise.](#)

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. [Voir Informations légales.](#)

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. [See Legal notice.](#)

Download PDF: 21.12.2024

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

Ein-Chip-Signalprozessoren in der Mess- und Regelungstechnik

H. Hanselmann

Der Aufsatz setzt sich mit der Leistungsfähigkeit moderner Signalprozessoren in mess- und regeltechnischen Anwendungen auseinander, wobei insbesondere Hard- und Softwareprobleme sowie die Möglichkeiten für eine rationelle Entwicklung von Reglern besprochen werden.

Cet article concerne les possibilités de processeurs modernes de signaux pour emplois en métrologie et en régulation. Des problèmes relatifs au matériel et au logiciel sont discutés, ainsi que les possibilités d'un développement rationnel des régulateurs.

Dieser Aufsatz ist eine überarbeitete Fassung des Vortrags, den der Autor an der Informationstagung «Digitale Signalprozessoren in der Informationstechnik» der Informationstechnischen Gesellschaft des SEV und des IEEE Switzerland Charter on Digital Communication Systems am 7. November 1985 gehalten hat.

Adresse des Autors

Dr.-Ing. H. Hanselmann, Universität-GH-Paderborn, Fachbereich 10, Maschinentechnik, Pohlweg 55, D-4790 Paderborn.

1. Einleitung

Digitale Ein-Chip-Signalprozessoren (DSP) sind Ende der 70er Jahre aus der Erkenntnis heraus auf den Markt gekommen, dass viele Aufgaben der digitalen Signalverarbeitung mit universellen Mikroprozessoren (UMP) nicht oder nicht rationell gelöst werden können. Sowohl die UMP wie auch die DSP haben sich seither weiterentwickelt, und die Leistungsspanne zwischen beiden Gruppen hat sich dabei erhalten. Grob gesprochen ist mit DSP in der digitalen Signalverarbeitung mehr als eine Größenordnung höhere Rechenleistung möglich als mit vergleichbaren UMP. Der Grund liegt hauptsächlich in der speziellen Architektur der DSP.

2. DSP-Architektur

- Hauptmerkmale heutiger DSP sind
- schneller 16·16-bit-Hardware-Multiplizierer on-chip (100...300 ns),
 - On-Chip-Datenspeicher (128...544 Worte à 16 bit),
 - Doppelwort-Akkumulator,
 - Barrel-Shifter,
 - zahlreiche interne Datenpfade,
 - Festkomma-Arithmetik.

Die genannten Merkmale findet man (mit kleinen Einschränkungen) bei allen DSP, die zur Zeit für den allgemeinen Einsatz in Frage kommen. Die Tabelle I zeigt eine Aufstellung dieser Prozessoren, wobei nicht alle-

mein zugängliche oder nur maskenprogrammierbare DSP nicht aufgeführt sind.

Die Rechenleistung schöpfen die DSP vor allem aus dem Hardware-Multiplizierer, aber auch aus der mehr oder weniger optimalen Verknüpfung aller wesentlichen Prozessorkomponenten über entsprechende Datenpfade. Typisch ist eine Zweiteilung des Datenspeichers mit zwei getrennten Datenpfaden, durch die paralleles Einspeisen von Operanden in den Multiplizierer und in die ALU möglich wird. Dabei kann ein ALU-Operand z.B. noch einer Multi-Bit-Verschiebung in einem Schritt über eine Verschiebe-Hardware (Barrel-Shifter) unterzogen werden. Der schnelle Multiplizierer und seine effiziente Ankopplung sind die Basis für die schnelle Ausführung von Skalarproduktoperationen, mit der Elementaroperation

$$acc := acc + x_i \cdot y_i \quad (1)$$

Die Durchsatzraten heutiger DSP liegen zwischen 2,5 und 10 Mio Elementaroperationen pro Sekunde.

Die in Tabelle I angegebenen DSP sind für 16-bit-Wortbreite ausgelegt. Entsprechend breit sind auch die Datenpfade vom und zum Speicher. Die Komponenten im Prozessorkern sowie ihre Verbindungen sind zum Teil breiter. Von Interesse ist insbesondere die Breite der Akkumulation. Zwei vorzeichenbehaftete 16-bit-Operanden ergeben ein 31-bit-Produkt. Um nicht frühzeitig Genauigkeit zu verlieren, sollten ALU und Akkumulator mindestens ebenso breit sein. Bei den DSP aus Tabelle I findet man Akkumulatorbreiten von 2·16 (geteilter Akkumulator), 26, 32 und 35 bit. Breiten kleiner 31 bit verhindern praktisch Rechnungen mit erhöhter Genauigkeit. Größere Breiten erleichtern auch die Behandlung von arithmetischen Überläufen.

Ein-Chip-Signalprozessoren

Tabelle I

Hersteller	Typ
Nippon Electric	7720
Texas Instruments	TMS 32010
Fujitsu	MB 8764
Standard Teleph. & Cable	DSP 128
Texas Instruments	TMS 32020

Um den Kern herum findet man bei modernen DSP inzwischen auch Einrichtungen zur automatischen Befehls- oder Befehlssequenzwiederholung, zur externen Speichererweiterung, zur Kommunikation, zum Multiprozessorbetrieb und zur effizienten Speicheradressierung. Letzteres geht schon bis zur «Registerrelativen Autoinkrement modulo Vektorlänge»-Adressierung (DSP 128). Diese Adressierungsart beschleunigt beispielsweise die Durchrechnung von FIR(Finite Impulse Response-)Filtern. Dort müssen die Elemente des Koeffizienten- und des Datenvektors zyklisch adressiert werden. Gleichzeitig ist von Abtastschritt zu Abtastschritt zur Realisierung der Verschiebeoperation der Anfang im Datenvektor um eine Position zu versetzen.

3. Einsatzgebiet Messtechnik

In der Messtechnik können folgende Anwendungen als typisch angesehen werden

- lineare digitale Filterung (IIR, FIR),
- Spektralanalyse über DFT bzw. FFT,
- nichtlineare Operationen auf Messdaten,
- Messgeberlinearisierungen,
- Kalman-Filterung,
- Signalgenerierung (Sinus, Rauschen, usw.).

Folgende Resultate, die für den TMS 32010 Gültigkeit haben, zeigen die Leistung von DSP in der Messtechnik; Bearbeitung von IIR-Filtern in 2,2 μ s pro Sektion 2. Ordnung und von FIR-Filtern in 0,4 μ s pro Anzapfung; komplexe 64-Punkte-FFT mit laufzeit-optimaler Codierung in 0,6 ms, bei 1024 Punkten in 43 ms [1]; Table-Lookup für eine nichtlineare Kennlinie inklusive linearer Interpolation in 8 μ s; Erzeugung von PRBS-Rauschen (Maximalfolge) aus einem 32-bit-Register mit 5,4 μ s pro Takt.

Die hohe Rechenleistung der DSP erlaubt in vielen Fällen, vor Ort intelligente Datenreduktion durchzuführen. In einem Mehrkanal-AD-Wandlersystem mit direkt angekoppelten DSP kann dieser etwa Aufgaben wie die Behandlung komplizierter Triggerbedingungen, Mittelwertbildung (moving average) oder Datenratenreduktion durch Filterung übernehmen. Ein kommerzielles Produkt in dieser Richtung gibt es bereits; eine VME-Bus-

Karte mit einem TMS 32010 für 4 AD-Kanäle.

4. Einsatzgebiet Regelungstechnik

In der Regelungstechnik sind neben dem eventuellen messtechnischen Teil DSP vor allem dort von Interesse, wo hohe Abtastfrequenzen bei gleichzeitig hoher Reglerordnung verlangt, und eventuell noch komplizierte nichtlineare Berechnungen in jedem Abtastschritt nötig sind. Diese Situation findet man bei

- schnellen elektromechanischen Positionierungen,
 - aktiver Schwingungsdämpfung,
 - Servohydraulik,
 - Roboterregelung,
 - Antriebstechnik,
- insbesondere aber beim Einsatz komplexerer Regler, etwa bei Zustandsreglern mit Beobachtern oder Kalmanfiltern [2] oder auch bei anders ausgelegten und strukturierten Mehrgrößenreglern [3].

4.1 Hohe Abtastfrequenzen

Ein Beispiel für sehr hohe Abtastfrequenzen (10...30 kHz) ist die Spurlage- regelung in Magnetplattenspeicher- Laufwerken. Der elektromagnetische Antrieb des Kopfpositionierers wird bei höheren Speicherdichten über Servospuren in einem geschlossenen Spurlageregelkreis betrieben. Dabei werden Regelbandbreiten >1 kHz angestrebt. Wegen zahlreicher Struktur- schwingungsresonanzen und Tilgungen (Einkerbungen im Betrags- frequenzgang) in der Regelstrecke im Bereich einiger 100 Hz bis über 10 kHz ist gleichzeitig ein erhöhter Aufwand im Regler mit Reglerordnungen zwischen

Erreichbare Abtastfrequenzen

Tabelle II

Mikroprozessor	Clock	f_a
8086	8 MHz	< 2 kHz
Z8000	5 MHz	< 2 kHz
MC 68000	10 MHz	< 4 kHz
NS 32016	10 MHz	< 5 kHz
TMS-32010-Signalprozessor		31 kHz

5 und 10 nötig. Die Tabelle II zeigt für einen solchen Fall, dass die Rechenleistung von universellen Mikroprozessoren (UMP) hier bei weitem nicht ausreichend ist. Die hier angegebenen Abtastfrequenzen f_a für die UMP sind abgeschätzte Obergrenzen. Sie basieren auf den Zeiten für die UMP deutlich langsamste Operation, die 16·16-bit-Multiplikation. Weitere Operationen, wie Additionen und Datentransport, reduzieren die tatsächlich erreichbaren Abtastfrequenzen auf etwa 50 bis 70%, je nach Programmierstil (Unterprogramme und Schleifen oder «straight code»). An einem MC-68000-System wurden diese Verhältnisse auch verifiziert.

Der Vergleich basiert, auf einem Regler 9. Ordnung (Fig. 1) mit einem Eingang (Spurlagefehler) und einem Ausgang (Antriebsspulenstrom), der auf 33 Multiplikationen und 27 Additionen pro Abtastschritt führt. Die relativ hohe Reglerordnung kommt zustande, weil die Regelstrecke im interessanten Frequenzbereich zahlreiche komplexe Pole in der Übertragungsfunktion hat. Zur Kompensation dieser Resonanzen wurden Filter (sog. Notch-Filter) eingesetzt. Pro Resonanzstelle erhöht sich dadurch die Reglerordnung um 2. Gleichzeitig wurden Betragsanhebungen dort vorge-

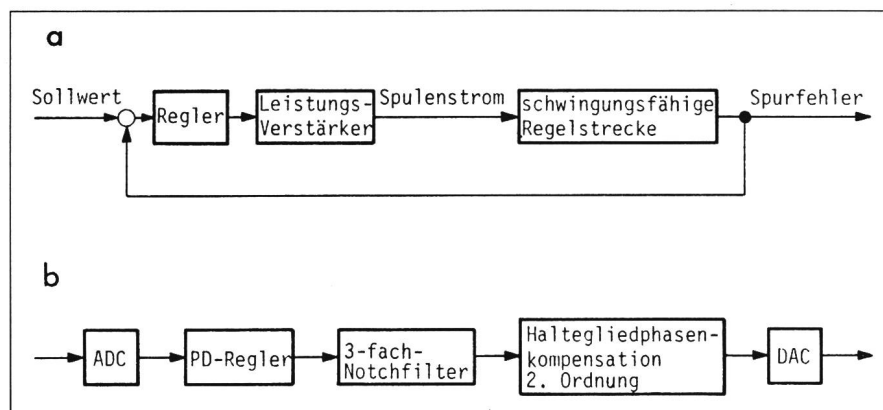


Fig. 1 Beispiel Spurlage- regelung

a Regelkreis b Regler

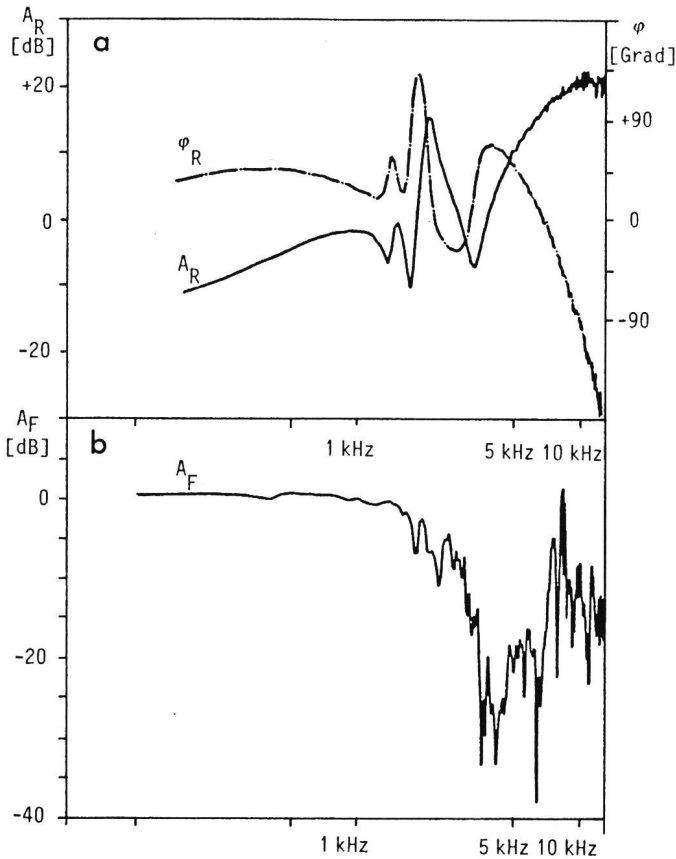


Fig. 2
Beispiel
Spurlageregelung für
Magnetplattenlaufwerk
 a Reglerfrequenzgang
 A_R Betrag
 φ_R Phase
 b Führungsfrequenzgang
 A_F Betrag

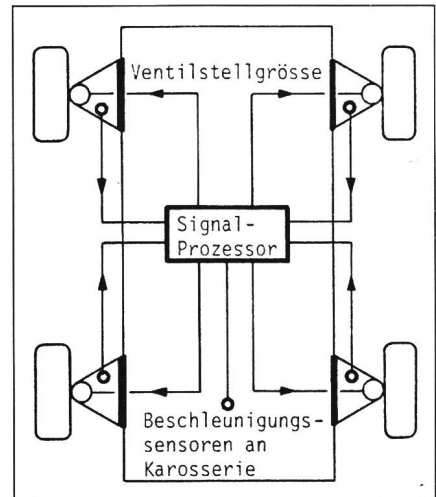


Fig. 3 Aktive oder semiaktive Fahrzeugfederung

- Sensoren
- Ventile

nommen, wo die Regelstrecke Einkerbungen im Betragsfrequenzgang hat, allerdings ohne dabei die Reglerordnung weiter zu erhöhen. Den resultierenden gemessenen Frequenzgang des Reglers alleine zeigt die Figur 2a, den gemessenen Führungs-Betragsfrequenzgang des geschlossenen Regelkreises Figur 2b. Die Bandbreite ist etwas grösser als 1 kHz. Die Spitze bei 8 kHz wurde durch höherfrequente Strukturschwingungen in der Regelstrecke verursacht und später durch ein einfaches analoges Anti-Aliasing-Filter vor dem AD-Wandler beseitigt.

Eine ganz ähnliche Situation wie bei der Spurlageregelung ist in [4] für eine Plattformregelung beschrieben. Die Abtastfrequenz muss dort nicht so hoch sein, dafür ist aber die Gesamtordnung des sich aus mehreren Teilen zusammensetzenden Reglers noch höher.

4.2 Komplizierte Regler

Bei den oben zitierten Beispielen handelt es sich vom Ansatz her um durchaus konventionelle Regler, bei denen man gewöhnlich einfachere Verhältnisse erwartet. Hohe Reglerordnung und zusätzlich grössere Ansprüche an die Rechengenauigkeit erwartet man bei Zustandsreglern oder

anderen vergleichbaren Mehrgrössenreglern; bei Zustandsreglern vor allem deshalb, weil sich die Ordnungen der Modelle der Regelstrecke und der Anregungssignale (Führung und Störung) kumulativ im Beobachter oder Kalmanfilter niederschlagen, jedenfalls solange man keine Ordnungsreduktion vornimmt.

Ein Beispiel für eine relativ anspruchsvolle Mehrgrössenregelung ist die aktive Schwingungsdämpfung von Fahrzeugen [3; 5; 6], bei der die passiven Feder-Dämpfer-Elemente durch geregelte Hydrauliksysteme ersetzt werden, entweder mit Hydraulikversorgung (aktive Federung) oder ohne (semiaktive Federung, variabler Stossdämpfer). Die geforderten Abtastfrequenzen sind zwar nicht besonders hoch. Man kann Abtastfrequenzen zwischen 0,5 kHz (Regelung der langsameren Aufbaubewegungen) und 5 kHz (schnelle Hydraulik) vorschlagen. Jedoch gibt es pro Abtastschritt einiges zu rechnen, wenn 4 Halbachsen gleichzeitig geregelt werden müssen. Es handelt sich um einen echten Mehrgrössenregler mit 4 Stell- und mehr als 10 Messgrössen (Fig. 3). Die Reglerordnung wird insgesamt im Bereich 10 bis 30 liegen. Ausserdem sind in jedem Abtastschritt erhebliche

nichtlineare Berechnungen zur Kompensation nichtlinearer Effekte in der Regelstrecke auszuführen.

Mehrgrössenregler sind auch bei der elektronischen Dämpfung und Versteifung elastischer Roboter [7; 8; 9] gefordert. Die Strategie ist, Schwingungseffekte in einer unteren Regelungsschicht aktiv zu dämpfen und der darüberliegenden Bahnregelungsschicht in etwa ein Starrkörpersystem zu präsentieren. Zahlreiche Messgrössen (Lage, Geschwindigkeit, Armverbiegung, für jede Achse) müssen zu den Motorstellgrössen verknüpft werden. Dabei spielen u. a. Reibungsbeobachtung und -kompensation eine Rolle. Die Figur 4 zeigt beispielhaft den gemessenen Verlauf der Krümmung eines Roboterarms, einmal mit und einmal ohne aktive Schwingungsdämpfung, bei ansonsten gleicher Schnelligkeit. Zur Realisierung der bei den Untersuchungen nötigen Regelun-

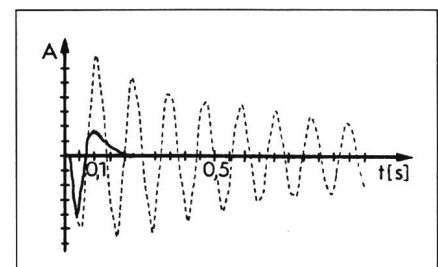


Fig. 4 Schwingungsdämpfung an einem Roboter [9]

- A Armverbiegung bei Lagesollwertsprung
- t Zeit
- übliche Starrkörperregelung
- aktive Schwingungsdämpfung

gen wurde und wird, wie auch bei den Experimenten zur Plattenlaufwerksregelung und zur aktiven Federung, ein TMS-32010-Laborsystem eingesetzt.

4.3 Nichtlineare Systeme

In den beschriebenen Anwendungen wird der DSP im wesentlichen zur Realisierung *linearer* zeitinvarianter Mehrgrößenregler benutzt, zum Teil mit konzentrierten nichtlinearen Erweiterungen. Ein andersartiges Anwendungsbeispiel findet sich in der elektrischen Antriebstechnik bei der feldorientierten Regelung von Drehstrom-Asynchronmotoren, wo u.a. eine Reihe *nichtlinearer* Berechnungen in jedem Abtastschritt zur Bestimmung des Magnetflusses aus Statorströmen und Drehzahl eine hohe Rechenleistung verlangen [10].

5. Probleme mit DSP

5.1 Hardware

Hemmend für den DSP-Einsatz im industriellen Bereich dürfte der im Vergleich zu Mikrocontrollern relativ grosse Zusatzaufwand an Hardware um den DSP herum sein, d.h. der Aufwand an schnellen Speichern – teilweise mit Zugriffszeiten < 50 ns – und Peripheriebausteinen. Mikrocontroller wie der Intel 8096 oder der NEC 78312 enthalten auf dem Chip bereits AD-Wandler, Pulsbreitenmodulatoren, Zustandswechseldetektoren, Zähler, asynchrone Schnittstellen bis hin zur kompletten Winkeldecodierlogik für inkrementale Winkelgeber. Andererseits erbringen Mikrocontroller kaum mehr Rechenleistung als universelle Mikroprozessoren. Eine interessante Lösung ergibt die Kombination eines DSP mit einem Mikrocontroller. So wird beispielsweise bei einer antriebstechnischen Anwendung in der Industrie ein Intel-8096-Mikrocontroller mit einem TMS 32010 gekoppelt. Auch die Kopplung mit einem UMP kann interessant sein, wie die Lösung bei einer Kreisel-Navigations-Anwendung mit einem MC 68000 als UMP und einem TMS 32010 als DSP zeigt. Ein anderes Beispiel ist das in [11] beschriebene Multiprozessorsystem für Roboterregelungsaufgaben.

In Anwendungen mit mehreren asynchronen Ereignissen, auf die vom Prozessor reagiert werden muss, kann die zum Teil völlig fehlende, in jedem

Fall aber stark beschränkte Interruptfähigkeit stören. Weitere Einschränkungen bringt der teilweise kleine lokale Datenspeicher (stark verbessert bei neueren DSP), der zu externen und weniger effizienten Speichererweiterungen oder zu anderen verlangsamenden Lösungen [12] zwingen kann, sowie die Beschränkung auf Festkommaarithmetik. Gleitkommaarithmetik ist mit den in Tabelle I genannten DSP nur zum Teil möglich und dann mit Rechenleistungen, die nicht über denen gängiger UMP-Gleitkomma-Prozessorkombinationen liegen. Die meisten Anwendungen kommen mit Festkommaarithmetik auch gut aus, sofern die Vorbereitung des zu realisierenden Systems (Regler, Filter) sorgfältig erfolgt (siehe Abschnitt 7).

5.2 Software

Nicht zuletzt muss auch die eigentliche Programmierung der DSP als nicht immer unproblematisch angesehen werden. Dafür sorgt zunächst einmal schon die spezielle Prozessorarchitektur. Sie reflektiert sich in Befehlssätzen, die im Vergleich zu denen moderner UMP zum Teil recht ungewöhnlich aussehen, neuartige Konstruktionen enthalten und andere gewohnte z.T. vermissen lassen (z.B. sind Einzelbit-Befehle teilweise nicht vorhanden). An bestimmten Stellen muss bei manchen DSP auf Pipelineeffekte Rücksicht genommen werden. Beim MB 8764 beispielsweise kann man alle 100 ns (Befehlszykluszeit) ein Produkt aus dem Multiplizierer beziehen, es gehört aber zu Operanden, die zwei Zyklen vorher eingespeist wurden. Oft sind Befehle in ihrer Anwendbarkeit beschränkt. Eine Befehlskomponente für die Verschiebung eines Datenwortes während eines Datentransportes etwa kann in einem Befehl für in Grenzen beliebige Verschiebungen gelten, in einem anderen Befehl aber auf Verschiebungen um 0,1 oder 4 bit eingeschränkt sein.

Derartige Unschönheiten und andere Einschränkungen führen nicht selten zu ziemlich langen Codekonstruktionen für relativ einfache Aufgaben. Der Effizienz wegen sind solche Konstruktionen oft auch noch stark an das jeweils vorliegende Zahlenmaterial gebunden. Ein Beispiel ist das Abspeichern eines Skalarproduktresultats, das mit herunterskalierten Koeffizienten berechnet wurde. Sei

$$r = c^T x \quad (2)$$

das Original-Skalarprodukt¹, das aber wegen betragsmässig zu grosser Koeffizienten in c^T mit um einen Zweierpotenzfaktor herunterskalierten Koeffizienten

$$c_s^T = \frac{1}{2^v} c^T \quad (3)$$

im Prozessor berechnet wird. Das Resultat muss dann wertmässig vor der Weiterverwendung mit 2^v multiziert werden, was durch um v bit versetztes Abgreifen des Resultats aus dem Akkumulator erreicht wird (Rückskalierung). Diese Aufgabe tritt häufig bei Reglern auf (siehe unten) und muss unter Überlaufüberwachung und -behandlung erfolgen. Resultate, die bei der Rückskalierung überlaufen würden, müssen unbedingt auf den vorzeichenrichtigen Maximalwert gesetzt werden (Sättigung). Ein in [13] gezeigtes Programm für den TMS 32010 leistet dies für alle sinnvollen Skalierungsfaktoren, benötigt aber zwischen 13 und 20 Befehlszyklen. Bei kurzen Skalarprodukten steckt hier mehr Aufwand als im Skalarprodukt selbst. Ein anderes Programm kommt zwar mit 6 bis 10 Zyklen aus, funktioniert aber nur für $v = 0$ und 3.

6. Programmierunterstützung

6.1 Assembler

Von den DSP-Herstellern selbst wird bisher nur Makro-Assembler-Unterstützung angeboten. Der Wunsch, einen besonders effizienten Code zu schreiben, führte viele Anwender bisher ohnehin zur Assemblerprogrammierung. Das ist aber oft nicht gerechtfertigt, denn es gibt durchaus effiziente Programmierunterstützung oberhalb der Assemblerebene. Diese kann man in drei Klassen einteilen:

- Hochsprachencompiler (mit Einschränkungen und Erweiterungen),
- spezielle Signalverarbeitungscompiler,
- automatische Codegeneratoren.

6.2 Hochsprachencompiler

Hochsprachencompiler existieren kommerziell neuerdings für die Sprachen C und Pascal beim TMS 32010

¹ Der Buchstabe T in (2) bedeutet «transponiert».

[14]. Für den C-Compiler wird eine Effizienz von 75% relativ zur Assemblerprogrammierung angegeben; für den Pascal-Compiler ist eine derartige Angabe nicht bekannt. Die Effizienz könnte aber ähnlich liegen, wenn Sprachkonstrukte vermieden werden, die nur mit umständlichen Assembler-Befehlskonstruktionen auf den DSP abgebildet werden können. Der Sprachumfang erscheint sehr komplett und wurde z. B. um den in der Signalverarbeitung gebräuchlichen Datentyp «fractional» (Festkommazahlen mit Werten zwischen -1 und +1) und um Interruptanbindung erweitert.

Wiederkehrende Unteraufgaben wie etwa digitale Filterung können mit nichtspezialisierten Hochsprachen wie C oder Pascal als Unterprogramme oder Module formuliert werden. Bei geeigneter Bezeichnerwahl (Namen von Variablen und Prozeduren) kann ein DSP-Programm dann einer abstrakten Formulierung der Signalverarbeitungsaufgabe ähneln.

6.3 Spezielle Signalverarbeitungscompiler

Ein anderer Ansatz ist die Definition einer speziellen signalverarbeitungsorientierten Sprache und der Bau eines Compilers zur Umsetzung in den Prozessorcode. Auch hierfür gibt es schon realisierte Beispiele [14; 15].

6.4 Automatische Codegeneratoren

Die automatischen Codegeneratoren schliesslich lösen das Programmierproblem für jeweils eine bestimmte Klasse von Aufgaben, wie z. B. digitale Filterung [16; 17] oder lineare zeitinvariante Mehrgrössenregler mit nichtlinearen Erweiterungen [13; 18], gegeben durch

$$x_{k+1} = A x_k + B u_k + f_x(x_k, u_k, y_k, k) \quad (4a)$$

$$y_k = C x_k + D u_k + f_y(x_k, u_k, k) \quad (4b)$$

Hier werden nicht Sprachmittel zur (fast) beliebigen Kombination bereitgestellt, sondern man geht von einer mehr oder weniger festen Programmarchitektur aus. Das tatsächliche DSP-Programm wird abhängig vom Zahlenmaterial und zusätzlichen Spezifikationen automatisch erzeugt. Je nach investiertem Aufwand bei der Erstellung des Generators können auch komplizierte Situationen beherrscht werden, ohne dass der Benutzer sich darum kümmern muss. Beispiele sind: die abgestufte Benutzung besonderer

Techniken, um mit dem vorhandenen Datenspeicher auszukommen, die Auswahl optimaler Befehlssequenzen für die Überlaufbehandlung oder die Einführung erhöhter arithmetischer Genauigkeit bei Bedarf. Der in [13] beschriebene Codegenerator bietet z. B. Doppelwortarithmetik bei nur vierfacher Rechenzeit, ein Faktor, der bei dem verwendeten DSP (TMS 32010) nicht leicht zu erzielen ist. Er wurde durch besondere Tricks und unter Verzicht auf volle doppelte Genauigkeit erreicht. Die im vorhergehenden Abschnitt erläuterte Überlaufbehandlung ist ein Punkt, der gerade bei Reglern wichtig ist. Zumindest die Stellgrössen als Ausgangsgrössen des Reglers können und sollen zeitweilig an ihre Anschläge gehen. Die Auswahl des jeweils effizientesten Codes für die Überlaufbehandlung überlässt man am besten einem Codegenerator.

Wichtig ist, dass ein Codegenerator nach übergeordneten Gesichtspunkten vorgehen und optimieren kann. Beispielsweise dadurch, dass er die jeweils laufzeiteffizientesten Codes generiert, solange genügend Daten- oder Programmspeicher verfügbar ist, dass er aber auch automatisch bereits vorgenommene Codierungen geeignet revidiert, wenn der Speicher knapp wird.

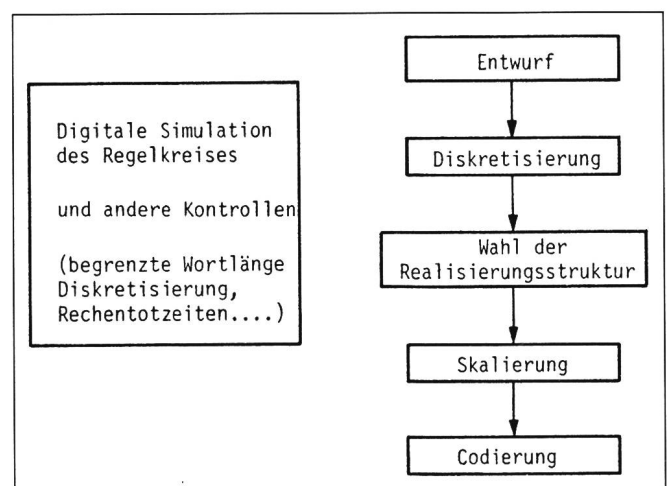
Generell kann ein Codegenerator stärker an die zu lösende Aufgabe angepasst werden als ein Hochsprachencompiler, weshalb sich relativ leicht ein effizienter Code erzielen lässt. Andererseits kann daraus eine geringere Flexibilität bei sich ändernder Aufgabe resultieren. Sinnvoll erscheint eine mehrstufige Zwischenlösung, bei welcher der Code nicht direkt aus den abstrakten Daten erzeugt wird, wie sie etwa die 4 Matrizen (*A*, *B*, *C*, *D*) des li-

nearnen Teils in (4) darstellen. Aus den Daten sollte vielmehr zuerst eine Formulierung der Aufgabe auf der Basis eines Satzes von Primitiven (ähnlich Funktionsaufrufen einer Hochsprache) gewonnen werden. Ein Beispiel für ein Primitiv wäre etwa der Aufruf eines Skalarprodukts, bei dem nicht nur die beiden Operandenvektoren und die Variable für das skalare Resultat, sondern auch Attribute spezifiziert würden. Solche Attribute betreffen Wortbreitenangaben, Skalierungsfaktoren oder die Forderung nach Abschneiden oder Rundung bei der Abspeicherung eines im Akkumulator stehenden Resultats. Die Primitiven müssen so aufgabennah gewählt sein, dass eine effiziente Umsetzung in den DSP-Code möglich ist. Nebenbei gewinnt man durch die Einführung der skizzierten Zwischenschicht eine grössere Unabhängigkeit vom Zielprozessor (verschiedene DSP oder auch UMP), weil die Softwarewerkzeuge oberhalb dieser Zwischenschicht dann nicht prozessorabhängig sind.

7. Schritte vor der Programmierung

Bevor ein System (Filter, Regler) für einen DSP programmiert wird, ist es immer ratsam und häufig zwingend, eine Reihe von vorbereitenden und kontrollierenden Schritten (Fig. 5) rechnergestützt zu durchlaufen [19]. Für den Fall eines analogen Reglerentwurfs müsste zunächst eine *Diskretisierung*, d. h. eine Überführung der Differentialgleichungen in Differenzgleichungen stattfinden. Dafür gibt es zahlreiche Verfahren unterschiedlicher Qualität [20]. Die nächsten Schritte, Strukturwahl und Skalierung, sind we-

Fig. 5
Vorgehen bei der
Reglerrealisierung



SIEMENS

Kennenlern-Aktion

SIMATIC-Komplettpaket zum Sonderpreis



Dies alles zusammen für nur
Fr. 1'200.— (exkl. WUST)

Die beste Gelegenheit, SIMATIC kennenzulernen – und die preisgünstigste dazu.

Da sollten Sie sofort zugreifen:
Karte ausfüllen und absenden.

- Steuerungsgerät
SIMATIC S5-101U mit
20 Ein- und 12 Ausgängen
- Programmiergerät PG 605U
- Simulator zum Programmtest
- Pufferbatterie

- Netzanschlusskabel
- Schraubendreher
- Programmierfibel
- Bedienungsanleitungen
für S5-101U und PG 605U

**Bestellkarte für das
SIMATIC-Komplettpaket**

Der Preisvorteil für Sie: Das SIMATIC-Komplettpaket für nur Fr. 1'200.- (exkl. WUST)

Wer speicherprogrammierbare Steuerungstechnik kennenlernen will, sollte es **jetzt** tun – mit SIMATIC, der Nr. 1 in Europa.

Und – im SIMATIC-Komplettpaket ist gleich alles dabei, was Sie zur Lösung einer Steuerungsaufgabe brauchen: Neben der Steuerung SIMATIC S 5-101U auch noch ein Programmiergerät (mit dem Sie auch grössere SIMATIC-Steuerungen programmieren können), ein Simulator (der Ihnen bei der Überprüfung des Programms hilft), eine Programmierfibel (damit der Umgang mit S 5-101U noch einfacher wird), eine Batterie (damit Ihnen auch nach dem Abschalten der Netzspannung Ihr Programm nicht „verlorengeht“), ein Schraubendreher, Netzanschlusskabel – und noch ausführliche Bedienungsanleitungen in deutscher Sprache.

Dies alles zusammen für nur Fr. 1'200.- (exkl. WUST)!

Automatisierungsgerät SIMATIC S5-101U

mit 20 Eingängen und 12 Relais-Ausgängen. S 5-101U ist ein anschlussfertiges, robustes Gerät im Kunststoffgehäuse für den starkstromnahen Einsatz.

Netzanschluss	AC 220 V/240 V, 50/60 Hz
Eingänge	20 (DC 24 V)
Ausgänge	12 (Relais), DC 24 V bzw. AC 240 V, jeweils 2 Ausgangskontakte mit gemeinsamem Potentialanschluss
Programmspeicher	1024 (1 · 2 ¹⁰) Anweisungen RAM intern, durch Batterie gepuffert
Programmbearb.-Zeit	70 ms/1024 (1 · 2 ¹⁰) Anweisungen
Funktionen	Binäre Verknüpfungen, Speicherfunktionen, Lade- und Transferfunktionen, Vergleichs- und Sprungfunktionen, Addition, Subtraktion
Zeiten	16 interne Zeiten 0,01 bis 9990 s
Zähler	16 interne Zähler 0 bis 999, davon 8 remanent
Merker	512, davon 256 remanent

Abmessungen 260 mm x 170 mm x 110 mm (B x H x T)

Gewicht ca. 2,7 kg

Handprogrammiergerät PG 605U
handlich, mit zweizeiliger Anzeige.

Simulator
zum Programmtest.

Batterie
Lithiumzelle für mindestens 1 Jahr Pufferzeit.

Netzanschlusskabel
2 m Länge.

Schraubendreher
mit isoliertem Schaft.

Programmierfibel
zum Selbststudium und zum Nachschlagen (74 Seiten)

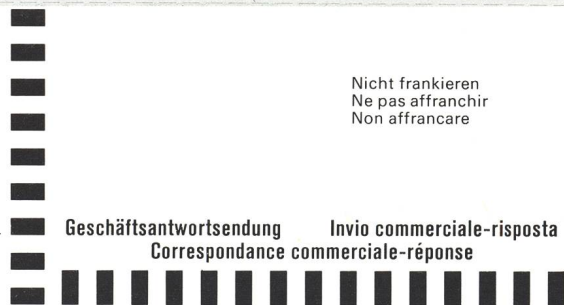
Bedienungsanleitungen
für S 5-101-U und PG 605U in deutscher Sprache in handlichen Kunststoffmappen.

Trennen Sie die Antwortkarte heraus und bestellen Sie Ihr SIMATIC-Komplettpaket! Bei einem solch günstigen Angebot werden Sie verstehen, dass mit einer Originalbestellkarte jeweils nur ein Paket abgerufen werden kann. Die Kennenlern-Aktion läuft bis zum 30. September 1986.

Siemens – Europas Nummer 1 bei speicherprogrammierbaren Steuerungen



Ich bestelle das SIMATIC-Komplettpaket. Den Betrag von Fr. 1'200.- (exkl. WUST) überweise ich nach Lieferung gemäss Rechnung.



Unterschrift _____

Senden Sie mir zunächst kostenloses Informationsmaterial über SIMATIC S 5-101U

Rufen Sie mich an und informieren Sie mich näher

Name _____

Firma _____

Branche _____

Adresse _____

Telefon _____

Siemens-Albis AG

Information 2
Freilagerstrasse 40

8047 Zürich

gen der Festkommaarithmetik unumgänglich. Zumindest die Strukturwahl fiele allerdings auch bei Gleitpunktarithmetik an.

Unter *Strukturwahl* fällt zum Beispiel die Transformation eines in Zustandsform geschriebenen Reglers oder Filters in eine andere Zustandsform mit besseren Eigenschaften hinsichtlich Koeffizienten- und Signalquantisierung. Der Regler wird intern anders strukturiert (anderes Zahlenmaterial, andere Besetzungsstruktur der Matrizen), ohne dass sich das Eingangs-/Ausgangsverhalten verändert. Nebenbei kann man u.a. eine Minimierung der Rechenzeit verursachenden Koeffizienten anstreben.

Die *Skalierung* ist nötig, um die intern auftretenden Variablen einerseits überlauffrei, andererseits aber wegen der Quantisierungseffekte gut ausgesteuert zu halten. Zusätzlich kann man mit der Skalierung die Koeffizienten in für die Arithmetik vernünftige Zahlenbereiche bringen. Bei der üblichen (und sinnvollen) Benutzung von «fractional»-Arithmetik mit Zahlenwerten im Bereich $-1,0$ bis $+1,0$ führt die Skalierung des Zustandsvektors bei Reglern allerdings oft auf viel zu grosse Zahlenwerte in der Ausgangsgleichung (4b). Dies liegt daran, dass Regler im Gegensatz zu Filtern oft Verstärkungen weit über 1 haben. Es sind dann besondere Massnahmen nötig, um einerseits mit den zu grossen Koeffizienten zurecht zu kommen, andererseits eine korrekte Überlaufbehandlung für die potentiell den Zahlenbereich überschreitenden Grössen zu garantieren (Abschnitt 5) und [13; 18; 19].

Führt man bei Festkommaarithmetik die Schritte Strukturwahl und Skalierung sorgfältig aus, so sind selbst Regler hoher Ordnung mit ungünstigem Zahlenmaterial oft überraschend gut realisierbar. Besondere Bedeutung hat neben anderen begleitenden Kontrollrechnungen die *Simulation* des kompletten Regelkreises, in der unbedingt die Prozessorarithmetik inklusive Überlaufverhalten, die AD- und DA-Wandlungseffekte sowie die zeitlichen Zusammenhänge (nicht gleichzeitiges Abtasten usw.) nachgebildet sein sollten [21]. Probleme in diesem Bereich werden dann vor der wirklichen Regelungsrealisierung erkannt und beseitigt. Erfahrungsgemäss ist

die Inbetriebnahme des Reglers an der Regelstrecke dann meist auf Anhieb erfolgreich. Allenfalls können noch Unzulänglichkeiten bei der Modellbildung der Regelstrecke, auf die sich der Regelungsentwurf stützt, für Überraschungen sorgen.

Attraktiv ist die vollständige Automatisierung aller Schritte zwischen dem Reglerentwurf und der Erprobung an der realen Regelstrecke. Dieses Ziel ist erreichbar. Ein guter Teil der nötigen Methodik existiert bereits. Beispielsweise sind mit den ursprünglich in [13; 18; 20; 21] beschriebenen, inzwischen weiterentwickelten Werkzeugen Reglerrealisierungen auf einem DSP schon in weniger als einer Stunde möglich, gerechnet vom Abschluss des Entwurfs bis zum ersten Einsatz an der realen Regelstrecke.

8. Weitere Entwicklung

Es zeichnet sich ab, dass DSP eine ähnlich stürmische Verbreitung wie vor Jahren die Mikroprozessoren finden werden. Zahlreiche Entwicklungen für den eigenen Bedarf der Firmen sind im Gange. Aber auch für den kommerziellen Markt sind eine ganze Reihe neuer Prozessoren in Sicht. In näherer Zukunft werden die meisten allgemein zugänglichen DSP bei Festkommaarithmetik bleiben. Bei den Bell Laboratories hat man jedoch schon einen 32-bit-Gleitkomma-DSP entwickelt. Mit der kompletten Arithmetikeinheit, 512·32-bit-Koeffizienten-ROM und 1024·32-bit-Daten-RAM auf dem Chip führt dieser DSP Gleitkommaoperationen so schnell aus wie die anderen DSP Festkommaoperationen. Auch NEC hat eine neue DSP-Familie angekündigt, wovon ein Mitglied ebenfalls für Gleitkommaoperationen ausgelegt ist [22].

Neben den universellen DSP werden auch kundenspezifische DSP-Bausteine Bedeutung erlangen. Sie werden entweder auf der Basis von DSP-Kernen entwickelt oder unter Verwendung parametrisierter (Wortbreite, Speichertiefe usw.) Makrozellen jeweils neu entworfen [23]. Das kann bei den zu erwartenden Stückzahlen in automobiltechnischen Anwendungen (aktive Federung) oder etwa bei der Magnetplattenlaufwerk-

regelung schon bald interessant sein. Know-how, das heute bei der Anwendung fertiger DSP erworben wird, kann morgen zum individuellen DSP führen.

Literatur

- [1] C.S. Burrus and T.W. Parks: DFT/FFT and convolution algorithms. Theory and implementation. New York, John Wiley, 1985.
- [2] J. Ulrich: Hinweise zur Anwendung des diskreten Zustandsreglers. Bull. SEV/VSE 76(1985)11, S. 630...636.
- [3] R. Kasper: Entwicklung und Erprobung eines instrumentellen Verfahrens zum Entwurf von Mehrgrößenregelungen. VDI-Fortschritt-Berichte. Reihe 8: Mess-, Steuerungs- und Regelungstechnik Nr. 90, 1985.
- [4] C. Slivinski and J. Borninski: Control system compensation and implementation with the TMS 32010. Texas Instruments digital signal processing application report. Dallas, Texas Instruments, 1985.
- [5] J. Lückel und R. Kasper: Mehrgrößenregelung. Entwurf und Realisierung moderner Mehrgrößenregelungen am Beispiel eines hydraulischen Fahrzeugprüfstands. Maschinenbau 13(1984)3, S. 13...16 + Nr. 4, S. 27...32.
- [6] F. Frühauf: Entwurf einer aktiven Fahrzeugfederung für zeitverschobene Anregungsprozesse. VDI Fortschritt-Berichte. Reihe 12: Verkehrstechnik Nr. 57, 1985.
- [7] H. Henrichfreise: Fast, elastic robots: control of an elastic robot axis accounting for nonlinear drive properties. Proceedings of the 11th IMACS World Congress, Oslo, 1985; vol. 4, p. 23...26.
- [8] W. Moritz, H. Henrichfreise and H. Siemensmeyer: A contribution to the control of elastic robots. IFAC Symposium on Robot Control, Barcelona 1985.
- [9] W. Moritz und H. Henrichfreise: Regelung eines elastischen Knickarm-Roboters. VDI/VDE-GMR Fachtagung für Steuerung und Regelung von Robotern, 12./13. Mai 1986, Langen b. Frankfurt.
- [10] P. Rojek und W. Wetzel: Mehrgrößenregelung mit Signalprozessoren. Elektronik 33(1984)16, S. 109...113.
- [11] W. Schumacher and W. Leonhard: Transistor-fed AC servo drive with microprocessor control. Proceedings of the International Power Electronics Conference, Tokyo 1983.
- [12] W. Loges: Regelsysteme höherer Ordnung mit dem Signalprozessor TMS 320. Elektronik 32(1983)25, S. 53...55.
- [13] W. Loges: Realisierung schneller digitaler Regler hoher Ordnung mit Signalprozessoren. VDI-Fortschritt-Berichte. Reihe 8: Mess-, Steuerungs- und Regelungstechnik Nr. 88, 1985.
- [14] K.E. Marrin: VLSI and software move DSP techniques into mainstream. Computer Design -(1985)September 15, p. 69...87.
- [15] O.E. Herrmann and J. Smit: A user-friendly environment to implement algorithms on single-chip digital signal processors. Proceedings of the European Association for Signal Processing -(1983), p. 851...854.
- [16] R. Schafer, R.M. Mersereau and T.P. Barnwell: Software package brings filter design to PCs. Computer Design -(1984)November, p. 119...124.
- [17] J. Skyttä a.o.: Experimental signal processing and development system. Proceedings of the European Conference on Circuits Theory and Design. Stuttgart. 1983. Berlin, VDE-Verlag, 1983; p. 290...292.
- [18] H. Hanselmann and W. Loges: Implementation of very fast state-space controllers using digital signal processors. Proceedings of the 9th IFAC World Congress 1984. New York, Pergamon-Press, 1984.
- [19] H. Hanselmann: Diskretisierung kontinuierlicher Regler. Regelungstechnik 32(1984)10, S. 326...334.
- [20] H. Hanselmann, R. Kasper and M. Lewe: Simulation of fast digital control systems. Proceedings of the first European simulation congress Aachen, September 12...16, 1983. Informatik Fachbericht Nr. 71. Berlin, Springer-Verlag, 1983.
- [21] P.R. Cappello: VLSI signal processing. New York, IEEE Press, 1984.
- [22] A. Zoicas: Signalprozessoren der zweiten Generation. Elektronik 35(1986)4, S. 59...68.