



Doctoral Thesis

Synthese und praktische Realisation von Systemen und Algorithmen für digitale Signalverarbeitung

Author(s):

Zeman, Jan V.

Publication Date:

1985

Permanent Link:

<https://doi.org/10.3929/ethz-a-000343676> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Synthese und praktische Realisation von Systemen und Algorithmen für digitale Signalverarbeitung

Abhandlung

zur Erlangung des Titels eines
Doktors der Technischen Wissenschaften
der

Eidgenössischen Technischen Hochschule Zürich

vorgelegt von

Jan V. Zeman

Dipl. El.-Ing. ETH
geboren am 3. Juni 1948
von Zürich

Angenommen auf Antrag von
Prof. Dr. G. S. Moschytz, Referent
Prof. Dr. W. Guggenbühl, Korreferent

Zürich 1985

Hartung-Gorre Verlag · Konstanz

KURZFASSUNG

Die vorliegende Arbeit befasst sich mit der Problematik der Echtzeit-Realisation von Signalverarbeitungsalgorithmen (SVA's) auf Rechnern mit grosser arithmetischer Parallelität, wie z.B. die Signal- und Array-Prozessoren (SP, AP). Die geleisteten Beiträge können wie folgt gegliedert werden:

- Entwicklung einer systematischen, datenflussorientierten Methodik zur Modellierung, Evaluation, Synthese und automatischen Programmierung von Signal- und Arrayprozessoren. Sie basiert auf einer virtuellen Implementierung eines gegebenen SVA auf einem Datenflussrechnermodell des Signalprozessors. Da nur SVA's mit konstantem Datenfluss behandelt werden, kann dieses Vorgehen für konventionelle, mikroprogrammgesteuerte Rechnerarchitekturen, als eine Art Datenflusscompilers wie folgt realisiert werden:

Zuerst wird der SVA als Datenflussgraph (Netzplan) dargestellt. Dann wird ein, für eine gegebene SP-Architektur zulässiger Datenflussablauf des SVA bestimmt, der gleichzeitig die SVA-Rechendauer und die Anforderungen an die SP-Ressourcen minimiert. Dies geschieht in zwei separaten Schritten: (i) der Schedule-Optimierung (Scheduling) und (ii) der Ressourcenzuordnung.

Der so erhaltene zeitliche Ablauf des Datenflusses gibt einerseits Aufschluss über die Interaktion SP-Architektur \leftrightarrow Algorithmus und dient somit als Mittel für eine interaktive SP-Architektursynthese.

Andererseits kann er anschliessend in ein Mikroprogramm umgewandelt werden, welches den Funktionsablauf des realen parallelen Signalprozessors steuert.

Auf diese Weise wirkt der SP wie ein Datenflussrechner, allerdings kann dabei der für das "real-time Scheduling und Resource Allocation" benötigte Hardware- und Ausführungszeit-Ballast eingespart werden.

Ein SVA mit konstantem Datenfluss kann somit in optimaler Weise (bezüglich der minimalen Rechendauer, Nutzung der SP-Ressourcen und inneren SVA-Parallelität) auf eine gegebene reale SP-Architektur übertragen und das dazugehörige Mikroprogramm automatisch generiert werden.

Gleichzeitig werden die für die Synthese der SP-Architektur wichtige Interaktion "Rechnerarchitektur \leftrightarrow Algorithmus" sowie die strukturelle Architektureffizienz technologieunabhängig erfasst. U.a. können folgende wichtige realisationstechnische Parameter einfach ermittelt werden: (i) die Länge des kritischen Pfades im SVA (T_{cp}), (ii) die benötigte Parallelität, Rechengeschwindigkeit und Pipelining der Recheneinheiten, um den SVA innert einer vorgegebenen Zeitlimite $T_g \geq T_{cp}$ ausführen zu können.

- Das Scheduling ermittelt den Zeitplan der Rechenoperationen unter Berücksichtigung der Vorgängerrelationen der Operationen und Ressourceneinschränkungen im SP-Modell (z.B. die Anzahl der Recheneinheiten, Busleitungen, Dauer der Rechenoperationen, Pipelining). Zur Lösung dieses generell NP-vollständigen Problems wurde eine neue, auf den Methoden der Netzplantechnik basierende, heuristische dynamisch-parallele Prozedur vorgeschlagen. Sie minimiert sehr zuverlässig die SVA-Rechendauer für die vorgegebenen Ressourceneinschränkungen, lässt aber offen in welchen konkreten Ressourcen die Operationen ausgeführt werden. In der Diskussion wird auch der Zusammenhang zwischen solchen heuristischen und den exakten Schedulingverfahren gezeigt.
- In der Ressourcenzuordnungsprozedur werden die Operationen und Zwischenresultate im Schedule den konkreten Recheneinheiten, Busverbindungen und Speichern des SP zugewiesen. Dank einer einfachen mathematischen Formulierung, konnte dieses Vorgehen als Branch-and-Bound Verfahren implementiert werden. Die Resultate der Ressourcenzuordnung definieren die benötigte Verbindungs- und Speicherstruktur des Signalprozessors; ihr Hardware-Aufwand kann für einen konkreten SVA gezielt minimiert werden.
- Basierend auf den obigen Beiträgen kann die interaktive Synthese einer Rechenarchitektur vorgenommen werden, die einem SVA optimal angepasst ist. Zu diesem Zweck wurde auch eine Familie von idealen Signalprozessor-Architekturprototypen (ISP) vorgeschlagen, welche sich sehr gut direkt für die obenerwähnte Realisation eignen und auch als Ausgangspunkt für Modellierung und Entwurf von weiteren SP-Architekturen dienen können.
- Die Methodik wurde praktisch als Programmpaket ANGAS, RAMPAS implementiert und ihre Anwendungsmöglichkeiten werden anhand von Beispielen der Digitalfilter- und FFT-Algorithmen illustriert.
- Die vorliegende Arbeit behandelt auch weitere Anwendungen der erarbeiteten Methodik und des dazugehörigen Programmpaketes, u.a. (i) systematische Ermittlung der SP-Grundparameter (Anzahl und Rechengeschwindigkeit der arithmetischen Einheiten, Pipelining, Bustransferkapazität etc.), (ii) Evaluation und Vergleich der Digitalfilteralgorithmen bezüglich ihrer inneren Parallelität, Rechendauer und Resourcepanforderungen, (iii) Aufteilung der Algorithmen bei gleichzeitiger Minimierung der Rechendauer und der Interprozessor-Kommunikation, (iv) Zerlegung der Vektoroperationen in maximal parallele Sequenzen, (v) Timing-Design für serielle Arithmetik.
- Aufgrund der Untersuchung der Parallelität und Quantisierungseffekte in den Digitalfilteralgorithmen sind auch Beiträge zum Thema der SVA-Synthese entstanden. Sie wurden in separaten Publikationen beschrieben: Block-State-Space Digitalfilterstrukturen in [ZemLindg79,81,81a] und die Untersuchung alternativer Multiplikationsmethoden in [ZeDüNa79, Zem85].

ABSTRACT.

In this dissertation, attention is focused onto the problem of real-time realization of signal processing algorithms on program-controlled computational structures with high degree of arithmetic parallelism, such as the signal and array processors (SP,AP).

The proposed systematic data-flow oriented method assists in modeling, architecture performance assessment and synthesis, and automatical programming of such structures.

The approach is based on the data-flow-graph-representation and virtual implementation of the algorithms on a high-level, data-flow-computer model of the signal processor.

For the signal processing algorithms with a constant data flow structure, the data-flow determined for the model can then be converted into a program controlling the signal processor. Hence, a conventional, program controlled, parallel computing structure emulates a data-flow machine, however without the otherwise necessary overhead for the scheduling and resource allocation (these tasks are performed in the compilation phase for the model).

This approach has the following advantages: (i) the algorithm can optimally be mapped onto a given or synthesized SP-structure, in the sense that the available SP-resources and inherent algorithm parallelism are fully utilized and the algorithm duration minimized, (ii) for the SP-architecture synthesis and evaluation, the interaction "computer architecture \leftrightarrow algorithm" can be assessed in terms of the structural architecture-efficiency (independent of current technology). Among others, the following key design parameters can be evaluated: (i) the length of the critical path in algorithm (T_{cp}), (ii) parallelism, speed and pipelining of arithmetic processing units necessary to complete an algorithm within a given time limit $T_g \geq T_{cp}$.

The mapping of an algorithm on a given architecture is carried out in two major steps: (i) the scheduling of arithmetic operations under constraints on number, speed and pipelining of arithmetic and bus resources, (ii) final resource allocation for a given schedule.

The scheduling employs graph-theoretical concepts and techniques from the activity network theory ("project management"). Based on these, a new robust heuristic dynamic-parallel procedure has been proposed to the solution of this generally NP-complete problem. It yields schedules of minimum duration obeying given resource constraints, without being specific in which resource a scheduled operation actually takes place. The general discussion covers also the link between the heuristic methods and the exact implicit branch-and-bound enumeration.

The subsequent resource allocation handles the mapping of scheduled operations and intermediate results onto arithmetic resources, memories and busses. The developed simple mathematical formulation of this NP-complete problem leads to a branch-and-bound-algorithm implementation. According to the chosen cost function, the resulting interconnection and storage structure can be optimized with respect to the number and size of memory units, busses and crossbar switches.

Hence, the above procedures assist in the interactive design of SP-architectures that are optimally adjusted to a given signal processing algorithm. To support this approach, a model of an ideal signal processor (ISP) has been proposed. It serves for the modeling of given SP-architectures and also as a starting point for the synthesis of special purpose devices for a specific algorithm.

The method has been implemented as a program package (ANGAS, RAMPAS). Its applications to the architecture-design and programming of signal processors are demonstrated on digital filters and FFT algorithms.

Among further applications of this methodology shown in this thesis are: (i) systematic determination of the arithmetic section specifications (number, speed and pipelining of arithmetic units), (ii) evaluation and comparison of digital filter algorithms with respect to their inherent parallelism, minimum possible computational duration and required parallel arithmetic resources, (iii) partitioning of algorithms, so that the computational time and the interprocessor communication minimized, (iv) partitioning of vector operations in parallel, time-balanced trees, (v) timing-design for serial arithmetic.

The author's investigations of the parallelism and quantization effects in the digital filter structures have also resulted in contributions to the solution of problems in the field of the synthesis of signal processing algorithms, namely: the block state-space structures and the alternative multiplication methods. These results have been published separately [ZemLindg79,81,81a], [ZeDüNa79, Zem85].