

Digit-Online Algorithmen und Architekturen

Steffen Dolling, Dirk Timmermann

Universität Rostock,
Fachbereich Elektrotechnik und Informationstechnik,
Institut für Angewandte Mikroelektronik und Datentechnik,
R.-Wagner-Str. 31, D-18119 Rostock,
email: dol@baltic.e-technik.uni-rostock.de

In dieser Arbeit werden Digit-Online-Verfahren und Architekturen betrachtet, die nach dem Most-Significant-Digit-First (MSDF)-Prinzip arbeiten. Diese Operationen, die auf einer redundanten Zahlendarstellung beruhen, benötigen ∂ (Online-Delay) anliegende Digits des Eingangsoperanden, bevor die erste Ziffer des Ergebnisses generiert werden kann. Für die Leistungsfähigkeit des Verfahrens ist dabei entscheidend, daß ∂ einerseits allgemein wesentlich kleiner als die Operandenlänge n ist und andererseits lediglich von der jeweiligen Operation und der Zahlenbasis r , aber nicht von n abhängt. Daraus resultieren die Vorteile der Digit-Online-Verfahren - eine serielle Kommunikation mit einer niedrigen Bandbreite und eine genauigkeitsunabhängige Latenzzeit. Da somit eine hohe Überlappung von aufeinanderfolgenden und möglicherweise voneinander abhängigen Operationen erreichbar wird, stellen diese Verfahren für stark kaskadierbare Anwendungen mit einem großem Rechen- und Kommunikationsaufwand eine Alternative zu parallel arbeitenden und kommunizierenden Strukturen dar.

Die Realisierung der Digit-Online-Verfahren erfolgt in intern parallel aufgebauten Rechenelementen, wobei die jeweiligen Algorithmen auf die Funktionalität generischer Grundzellen zurückgeführt werden. Im wesentlichen handelt es sich dabei um Multiplexer zur Realisierung der Multiplikation eines Digit-Vektors mit einem einzelnen Digit, Shiftstrukturen und Truncationen-Module zur Vermeidung von Pseudoüberläufen. Das zentrale Element bildet dabei eine chipflächenmäßig günstig realisierbare 4-zu-2 Signed-Digit-Addierervariante [1] und die entsprechenden 3-zu-2 und 2-zu-2-Derivate. Diese bilden auch die Basis für die notwendigen elementaren seriellen Grundzellen (z.B. Additions-Subtraktionszelle, Incrementer, 2-zu-2-Zelle). Die Abbildung der einzelnen Algorithmen auf diese Grundzellen ist einerseits durch eine induktive Methode möglich. Dabei werden die Operationsgleichungen und Strukturen für die parallele Abarbeitung durch Transformationen in serielle Architekturen überführt. Diese Variante ist insbesondere für einfachere Algorithmen (Addition, Multiplikation) geeignet. Andererseits können die jeweiligen Digit-Online-Funktionen nach einer speziellen Entwicklungsmethode [2] hergeleitet werden. Im wesentlichen werden dabei auf der Basis von Fehler- und Konvergenzkriterien Iterationsgleichungen für einen Partialrest abgeleitet, die mit Hilfe der genannten generischen Zellen realisierbar sind. Dabei wird zusätzlich eine Selektionsfunktion zur Bestimmung des Ergebnisdigits notwendig. Diese ebenfalls als generische Grundzelle abgebildete Struktur ist in unseren Entwürfen parametrierbar, wodurch der Ergebnisgenauigkeitsverlauf beeinflussbar wird.

Auf Grundlage der genannten Ausführungen wurden die Digit-Online-Algorithmen und Architekturen für die elementaren Funktionen (Addition, Multiplikation, Division, Quadratwurzel) abgeleitet und zur weiteren Simulation und Verifikation in VHDL-Verhaltens- und Strukturbeschreibungen überführt. Durch entsprechende Anpassungen bzw. Transformationen

dieser Algorithmen konnten Strukturen und die entsprechenden VHDL-Modelle zur Realisierung von zusammengesetzten oder Multioperand-Operationen erreicht werden. Diese weisen durch die genannte Zusammenfassung von Teiloperationen eine Verbesserung der Kostenfunktion bzgl. des Schaltungsaufwandes und des Online-Delay's gegenüber Rechenelementen auf, die durch eine Verkettung der Elementarfunktionen gebildet werden. Z.B. läßt sich das Online-Delay bei einer Addition von vier bzw. acht Signed-Digit-Operanden mit der Zahlenbasis $r=2$ von 4 auf 3 bzw. 6 auf 4 verringern. Weitere Beispiele in diesem Zusammenhang sind die Potenzfunktion, die Addition bzw. Multiplikation mit festen Operanden und die lineare Funktion.

Die Realisierung von komplexen Funktionen (Sinus, Cosinus, Logarithmus) erfolgte im Gegensatz zu der sonst üblichen Vorgehensweise (CORDIC-Funktion) auf der Basis einer Potenzreihenentwicklung. Der Vorteil dieser Lösung besteht darin, daß sich die komplexen Operationen durch eine modulare Verknüpfung der oben genannten Rechenelemente für zusammengesetzte bzw. Multioperand-Operationen realisieren lassen. Der Nachteil dieses Verfahrens liegt in der Verschlechterung des Verhältnisses zwischen dem benötigten Berechnungsaufwand und dem erreichten Genauigkeitsergebnis.

Bei den bisherigen Ausführungen wurde davon ausgegangen, daß die Operanden Fixed-Point-Werte in einem bestimmten Wertebereich darstellen. Bei einer stark kaskadierten Struktur kann durch die Veränderung der einzelnen Operanden im Datenfluß diese Annahme nicht aufrecht erhalten werden. Entsprechend wird die Einführung einer Floating-Point-Darstellung notwendig. Somit werden zusätzlich zu den aufgezeigten Online-Operatoren Strukturen z.B. zur Exponentenberechnung, zur Overflow-Kontrolle und zur Mantissen-Normalisierung notwendig.

Unsere derzeitigen Arbeiten konzentrieren sich entsprechend auf die Abbildung von Floating-Point-Rechenelementen, die Realisierung weiterer zusammengesetzter Funktionen und den Aufbau eines Demonstrations-Boards. Dieses besteht aus FPGA-Bausteinen zur Aufnahme der Online-Operatoren und einer Switch-Matrix für eine konfigurierbare Verknüpfung.

- [1] N. Takagi, H. Yasuura, S. Yajima, „High-Speed VLSI Multiplication Algorithms with a Redundant Binary Addition Tree“, IEEE Transaction on Computers., Vol. 9, S. 789-796, 1985
- [2] M.D. Ercegovic, T. Lang, "On-line arithmetic: a design methodology and applications in digital signal processing", VLSI Signal Processing, III, IEEE Press 1988, S. 252-263, 1988