Technische Informatik

Malte Baesler

FPGA Implementation of a Decimal Floating-Point Co-Processor with Accurate Scalar Product Unit

Shaker Verlag
Aachen 2012

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: Hamburg-Harburg, Techn. Univ., Diss., 2012

Printed in Germany.

ISBN 978-3-8440-1063-3
ISSN 1436-882X

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

FPGA Implementation of a Decimal Floating-Point Co-Processor with Accurate Scalar Product Unit (by Malte Baesler)

Scientific and engineering problems are usually modeled using real numbers but are solved on digital computers that approximate them by floating-point numbers applying the predominant standard IEEE 754-1985. Although we generally use and think in decimal numbers, this arithmetic is binary because the corresponding circuits require less area, data is stored more densely, and binary arithmetic is more suitable for scientific applications due to its higher performance and better error characteristic.

However, many decimal fractional numbers cannot be represented exactly in binary (e.g., 1/10=0.1 has no exact binary representation) and must be approximated introducing rounding errors. In numerous engineering, financial, and commercial applications these errors are not acceptable and may even violate legal requirements of accuracy. Hence, in 2008 the floating-point standard IEEE 754-2008 was approved that also incorporates specifications for a decimal arithmetic. Software libraries can implement this arithmetic but are usually 100 to 1000 times slower than equivalent floating-point operations in hardware. Therefore, fast hardware support for decimal arithmetic on modern computer architectures is desirable.

IEEE 754 floating-point arithmetic is well-conceived for elementary operations, causing least possible rounding errors. However, due to cancellation, more complex operations in numerical algorithms might introduce serious errors and can even raise the question whether the computed result solves the given problem or not. For instance, the scalar product is a widely used operation in numerical applications that is prone to cancellation. Hence, the implementation of the *accurate* scalar product in floating-point units can significantly increase the accuracy of many algorithms. Moreover, various scientific and engineering applications require informations about the quality of the computed result. Interval arithmetic offers a method to yield reliable results by computing guaranteed enclosures of real-valued expressions. Unfortunately, interval arithmetic is not supported well on modern floating-point units because switching the rounding mode requires many cycles. Therefore, an efficient hardware support for interval operations requires that rounding is inherent to each operation.

In the context of this thesis, new decimal fixed-point and floating-point algorithms were analyzed and a decimal floating-point co-processor has been implemented. The four elementary operations (addition, subtraction, multiplication, and division) as well as the accurate scalar product are supported. The arithmetic units are fully combinational and can be improved by a configurable number of pipeline registers with the exception of the decimal divider that works sequentially. The co-processor provides support for the data format *decimal64* and is fully compliant to IEEE 754-2008. Furthermore, the rounding mode is inherent to each operation allowing the implementation of an efficient interval arithmetic for reliable computing based on decimal floating-point arithmetic. Finally, the algorithms were optimized for FPGA architectures and have been implemented on a Xilinx Virtex-5 FPGA.

FPGA Implementierung eines dezimalen Gleitkomma-Coprozessors mit Unterstützung für das genaue Skalarprodukt (von Malte Baesler)

Wissenschaftliche und technische Probleme werden üblicher Weise mittels reeller Zahlen formuliert und anschließend auf digitalen Rechnern unter Verwendung des binären Standards IEEE 754-1985 gelöst. Obwohl wir normalerweise in dezimalen Zahlen denken und rechnen, hat sich aus historischen Gründen ein binärer Gleitkommastandard durchgesetzt, da die Komplexität der Schaltkreise kleiner, die Datenspeicherung kompakter und die Arithmetik auf Grund ihrer Geschwindigkeit und günstigen Fehlercharakteristik gut für wissenschaftliches Rechnen geeignet ist.

Viele dezimale Zahlen lassen sich jedoch nicht exakt als binäre Gleitkommazahl darstellen (wie z.B. 1/10=0.1) und müssen daher gerundet werden. Dabei entstehen Rundungsfehler, die in vielen Finanzanwendungen bzw. technischen Anwendungen nicht akzeptabel sind oder sogar gegen internationales Recht verstoßen können. Daher wurde im Jahre 2008 der neue Gleitkommastandard IEEE 754-2008 verabschiedet, welcher u.a. die Spezifikation einer dezimalen Arithmetik umfasst. Diese dezimale Arithmetik ließe sich durch Softwarebibliotheken realisieren, die entsprechenden Operationen wären jedoch 100 bis 1000 mal langsamer als vergleichbare Gleitkomma-operationen. Daher ist die Bereitstellung einer schnellen und hardwaregestützten dezimalen Arithmetik in modernen Prozessorarchitekturen wünschenswert.

Elementare Operationen werden gemäß IEEE 754 optimal, d.h. mit kleinstmöglichen Rundungsfehlern behaftet, ausgeführt. Bei komplexen, aus elementaren Operationen zusammengesetzten Algorithmen können jedoch aufgrund von Auslöschung große Fehler auftreten, sodass die Frage aufgeworfen werden kann, inwieweit das berechnete Ergebnis überhaupt noch das gegebene Problem löst. Beispielsweise ist das Skalarprodukt eine häufig verwendete Operation, die sehr anfällig für Auslöschung ist. Daher kann die Implementierung eines *genauen* Skalarprodukts die Qualität numerischer Algorithmen erheblich verbessern. Des Weiteren wird von vielen Anwendungen eine Aussage über die Genauigkeit des Resultats gefordert. Hierfür bietet sich die Intervallarithmetik an, welche das korrekte Ergebnis mittels einer oberen und unteren Grenze einschließt. Leider unterstützen modernen Gleitkommaeinheiten dies nur sehr schlecht, da das Umschalten des Rundungsmodus sehr viele zusätzliche Takte benötigt. Eine leistungsfähige Hardwareunterstützung muss daher den Rundungsmodus direkt mit den arithmetischen Operationen verknüpfen können.

In dieser Arbeit wurden neue dezimale Fest- und Gleitkomma-Algorithmen analysiert und ein dezimaler Gleitkomma-Coprozessor entwickelt, welcher die vier Grundrechenarten (Addition, Subtraktion, Multiplikation und Division) sowie das genaue Skalarprodukt unterstützt. Die arithmetischen Einheiten, mit Ausnahme des Dividierers, sind rein kombinatorisch, sodass der Datendurchsatz mittels konfigurierbarer Pipeline-Register erhöht werden kann. Alle Operationen sind konform zum IEEE 754-2008 Datenformat *decimal64*. Zusätzlich kann jede Operation mit einem entsprechende Rundungsmodus verknüpft werden, sodass die Implementierung einer effizienten dezimalen Intervallarithmetik bestmöglich unterstützt wird. Die dezimalen Algorithmen sowie der Coprozessor wurden für FPGAs, speziell für Xilinx Virtex-5 FPGAs, entwickelt und optimiert.