

Dynamische Rekonfiguration von arithmetischen Einheiten auf Bitebene

O. A. Pfänder and H.-J. Pfeiderer

Universität Ulm, Abt. Allgemeine Elektrotechnik und Mikroelektronik, Albert-Einstein-Allee 43, D-89081 Ulm, Germany

Zusammenfassung. Im Hinblick auf Flexibilität und Effizienz beim modernen Schaltungsentwurf lohnt es sich oft, arithmetische Basiseinheiten mit passend gewählter fixer Wortlänge so zu erweitern, daß sich in Zusammenschaltung mit anderen Elementen eine übergeordnete Funktionalität mit erweiterter Bitbreite erreichen läßt. In diesem Beitrag soll die Multiplikation als konkretes Anwendungsbeispiel herangezogen werden. Dabei liegt das Hauptaugenmerk darauf, wie mit geringem Aufwand aus einer grundlegenden starren Struktur ein dynamisch rekonfigurierbarer Multiplizierer hervorgehen kann, der sowohl für vorzeichenlose als auch vorzeichenbehaftete Zahlen in den wichtigsten Zahldarstellungen geeignet ist. Die Vorgehensweise basiert auf dem Einsatz von Multiplexern auf Bitebene.

1 Einführung

Rekonfigurierbare Hardware ist heute in verschiedenen Granularitäten bekannt, wovon sich einige der Varianten nicht nur einmalig, sondern auch zur Laufzeit konfigurieren lassen (Bondalapati and Prasanna, 2002). Auf diese Art sind etwa dynamisch veränderliche Multistandard- oder Multiprotokoll-Chips realisierbar, von denen man sich zukünftig erhofft, dass hohe Masken- und andere Herstellungskosten besser amortisiert werden können. Außerdem kann das Layout einer überschaubaren Grundeinheit, die als Einzelteil zur Zusammenstellung eines komplexeren Systems dient, weitaus besser optimiert werden als ein unhandlicher Monolith.

In vielen Einsatzgebieten wie z.B. bei der elektronischen Bildverarbeitung spielen komplexe arithmetische Operationen eine entscheidende Rolle und stellen hohe Anforderungen an Algorithmik und Hardware. Besonders häufig auftretende Operationen werden üblicherweise in die Hardwareseite verlagert (Oklobdzija, 2000), um die geforderte Rechengeschwindigkeit und den notwendigen Datendurchsatz erreichen zu können. Eine Multiplikation mit nicht konstanten

Faktoren gehört aus Sicht der Hardware-Entwicklung zu diesen komplexen arithmetischen Operationen und deren Realisierung kann sehr aufwendig und flächenintensiv werden. Obwohl in jüngster Zeit “Field-Programmable Gate Arrays” (FPGAs) immer leistungsfähiger werden und die Ressourcen zur zahlreichen und parallelen Implementierung solcher aufwendiger Rechenschritte bieten, werden große Multiplizierer immer häufiger auch als eingebettete Strukturen im FPGA realisiert, wie etwa der 18×18 bit-Multiplizierer im XILINX Virtex II Pro (Xilinx, 2004). Durch die Verlagerung der komplizierten Multiplikation in einen optimierten Baustein steht ein Großteil der konfigurierbaren Logikblöcke wieder für andere Aufgaben zur Verfügung (Haynes et al., 1999).

Bei eingebetteten Multiplizierer-Strukturen ist jedoch die Wahl der Wortbreite kritisch. Wenn diese zu hoch gewählt wird und der zu implementierende Algorithmus unter Umständen nur einen Bruchteil davon ausnutzt, sinkt die Effizienz. Andererseits darf sie auch nicht zu gering gewählt werden, um nicht von vornherein z. B. eine spätere Erhöhung der Genauigkeit unmöglich zu machen. Aus diesen Gründen erscheint es sinnvoll, die Wortbreite für die Multiplikation bewußt flexibel zu wählen, auch um wahlweise asymmetrische und symmetrische Strukturen realisieren zu können.

2 Konzept für Zusammenschaltung

2.1 Grundlegende Struktur und Eigenschaften

Die hier vorzustellende grundlegende Idee zur Realisierung der flexiblen Wortbreite bei der Multiplikation beruht darauf, einen übergeordneten Multiplizierer aus mehreren separaten Elementen zusammensetzen, die wiederum auch selbst individuelle und voll funktionsfähige Multiplizierer sind. Aus $m_1 \times m_2$ solchen Elementen mit jeweils $n_1 \times n_2$ bit Wortbreite kann somit eine Gesamtstruktur von $(m_1 \times n_1) \times (m_2 \times n_2)$ bit entstehen; siehe Abb. 1. Die wesentlichen Anforderungen an ein solches Multiplizierer-Element sind Gleichförmigkeit, Skalierbarkeit und die Ausstattung mit geeigneten Schnittstellen. Aufgrund seiner besonders regulären Struktur weist ein iterativer Multiplizierer im Parallel-Parallel-Aufbau mit Carry-Ripple-Technik nach Hwang (1979) von Anfang an

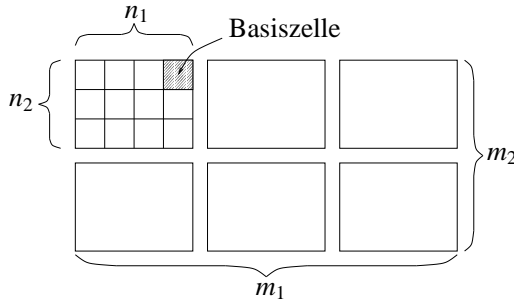


Abbildung 1. Eine $m_1 \times m_2$ Zusammenschaltung von $n_1 \times n_2$ Basiszellen.

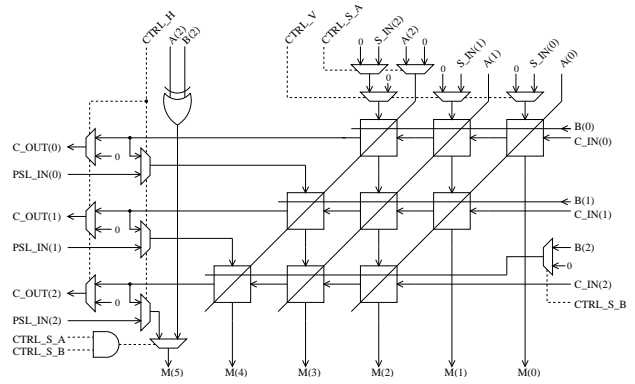


Abbildung 3. Konfigurierbarer Multiplizierer für "Signed-Magnitude" Zahlen.

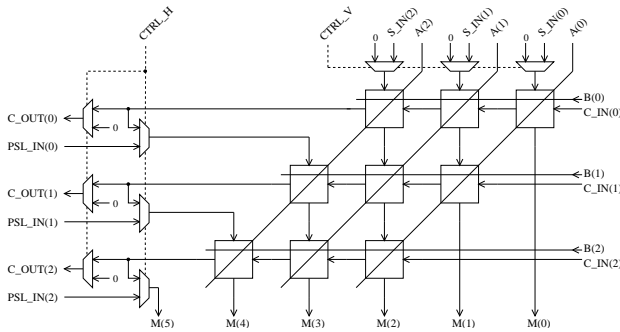


Abbildung 2. Konfigurierbarer Multiplizierer für vorzeichenlose Zahlen.

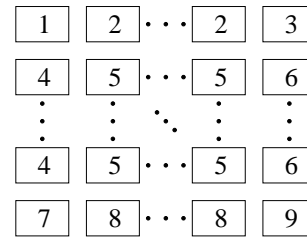


Abbildung 4. Neun verschiedene Positionen in einer zusammenschalteten Matrix.

die beiden zuerst genannten Eigenschaften auf und wird deswegen im folgenden genauer betrachtet. Die geforderte Zusammenschaltbarkeit kann mit nur wenig zusätzlichem Schaltungsaufwand erreicht werden, auch sind alle weiteren Maßnahmen zum Umgang mit vorzeichenbehafteten Zahlen stets Erweiterungen der grundlegenden Struktur mit einem unveränderten Kern.

Alle arithmetischen Basiszellen des hier eingesetzten Parallel-Parallel-Multiplizierers beinhalten je einen Volladdierer und ein UND-Gatter zur Berechnung des entsprechenden Partialprodukts. Da auch die Basiszellen in der oberen Reihe und der rechten Spalte identisch aufgebaut sind, existieren hier bereits zwei der geforderten Schnittstellen, die der Einspeisung eines n_1 bit breiten Partialsummensignals in die obere Reihe sowie n_2 zusätzlicher Carry-Bits von außerhalb dienen. Wie in Abb. 2 ersichtlich wird, können die Summeneingänge mit Hilfe von 1 bit-Multiplexern abgeschaltet werden. In ähnlicher Weise wird der Carry-Ausgangspfad auf der linken Seite der Matrix aufgetrennt: Eine doppelte Kette von Multiplexern sorgt entweder für die Rückführung der Signale in die jeweils darunterliegende Basiszelle oder für die Weiterleitung an ein benachbartes Multiplizierer-Element. Diese Multiplexer werden von zwei separaten Kontrollsignalen angesteuert, mit Hilfe derer sowohl die horizontale als auch die vertikale Schnittstelle des Elements rekonfiguriert werden können. Die Werte der Kontrollsignale sind abhängig von der Position des Elements in der

Zusammenschaltung (siehe auch Abb. 4) und können z. B. von einem Konfigurationsspeicher geliefert werden.

2.2 Vorzeichenbehaftete Zahlen

Das im vorigen Abschnitt beschriebene Multiplizierer-Element ist in seiner grundlegenden Form in der Lage, die Multiplikation mit vorzeichenlosen Zahlen durchzuführen. Es soll nun mit möglichst wenig Schaltungsaufwand so erweitert werden, daß es wahlweise auch mit vorzeichenbehafteten Zahlen umgehen kann.

Eine naheliegende Herangehensweise beruht auf der Wahl der "Signed-Magnitude"-Zahldarstellung nach Vorzeichen und Betrag. In dieser Darstellung beinhaltet das höchstwertige Bit einer n bit breiten Zahl das Vorzeichen und die übrigen $n-1$ bit ihren Betrag. Damit kann das Schema der Berechnung auf die vorzeichenlose Multiplikation zweier $n-1$ bit breiter Zahlen mit getrennter Verarbeitung des Vorzeichenbits zurückgeführt werden. Das Resultat ist negativ, wenn Multiplikand **A** und Multiplikator **B** verschiedene Vorzeichen haben. Die Vorzeichenbehandlung benötigt also eine XOR-Verknüpfung der beiden höchstwertigen Eingangsbits und kann entweder mit separater Logik außerhalb des Multiplizierer-Elements erfolgen oder auch mit Gattern, die ins Element integriert sind und zwei zusätzlichen Kontrollsignalen, wie es in Abb. 3 veranschaulicht wird.

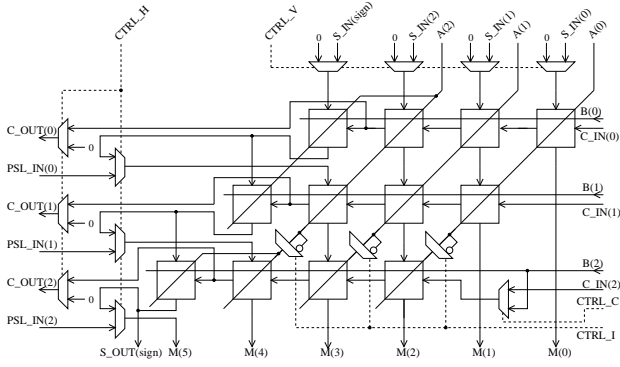


Abbildung 5. K2-Multiplizierer mit regulärer Vorzeichenerweiterung.

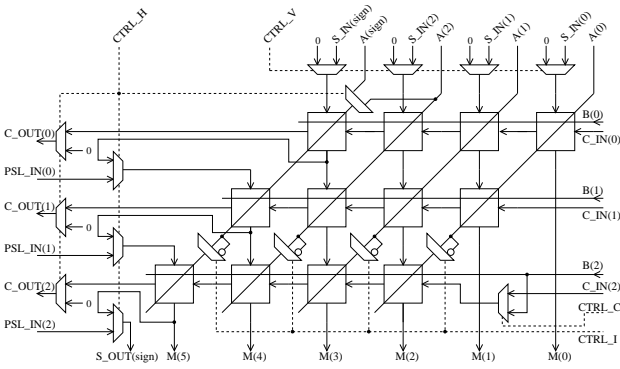


Abbildung 6. K2-Multiplizierer mit schaltbarer Vorzeichenerweiterung.

Wesentlich praxisrelevanter ist jedoch die Wahl der sogenannten K2-Zahldarstellung im Zweierkomplement, vor allem aufgrund der eindeutigen Repräsentierung der Null (Bermak et al., 1997). Um allerdings der geforderten Vorzeichenerweiterung bei der Wahl einer Parallel-Parallel-Struktur mit gleichförmigen Basiszellen gerecht zu werden (Parhami, 2000), ist dem Multiplizierer-Element eine zusätzliche Spalte an Basiszellen hinzuzufügen, vgl. Abb. 5. In einer Zusammenschaltung mehrerer Elemente zu einem übergeordneten Multiplizierer werden allerdings diese zusätzlichen Basiszellen ausschließlich in den Randpositionen 1, 4 und 7 benötigt und liegen in allen anderen Positionen brach. Wie in Abb. 6 gezeigt, werden aus diesem Grund zudem zwei Multiplexer eingeführt und die Verdrahtung der Extrazellen verändert, um im Endeffekt die Wortbreite des Multiplikanden **A** um 1 bit zu erhöhen. Das Resultat ist ein rekonfigurierbares $[(n_1+1) \times n_2]$ Multiplizierer-Element mit dynamisch schaltbarer Vorzeichenerweiterung.

Die Aufdoppelung des Vorzeichens kann aber durch geeignete mathematische Umformungen bei der Durchführung der Multiplikation vermieden werden. In der K2-Zahldarstellung wird dazu gemäß Baugh and Wooley (1973) und Bermak et al. (1997) der Wert des Produktes **M** zweier Zahlen **A** and **B** wie folgt ausgedrückt:

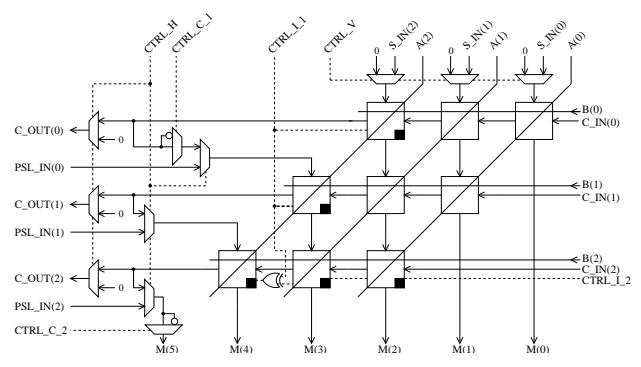


Abbildung 7. Modifizierter Baugh-Wooley K2-Multiplizierer mit schaltbaren Sonderzellen.

$$M_v = a_{n_1-1} b_{n_2-1} 2^{n_1+n_2-2} + \sum_{i=0}^{n_1-2} \sum_{j=0}^{n_2-2} a_i b_j 2^{i+j} + T_1 + T_2 \quad (1)$$

$$T_1 = 2^{n_1-1} \left(-2^{n_2} + 2^{n_2-1} + 1 + \sum_{j=0}^{n_2-2} \overline{a_{n_1-1} b_j} 2^j \right) \quad (2)$$

$$T_2 = 2^{n_2-1} \left(-2^{n_1} + 2^{n_1-1} + 1 + \sum_{i=0}^{n_1-2} \overline{a_i b_{n_2-1}} 2^i \right). \quad (3)$$

Diese Zerlegung des ursprünglichen Terms findet sich in einer modifizierten Multipliziererstruktur nach Baugh and Wooley (1973) wieder, die grafisch in Abb. 7 dargestellt ist. Die Basiszellen in der linken Spalte sowie der unteren Zeile der Matrix besitzen einen zusätzlichen internen Inverter. Dieser ist durch ein Kontrollsignal steuerbar und gleicht damit einem Multiplexer mit einem invertierenden Eingang, was der Funktion eines XOR-Gatters entspricht. In Abb. 7 wird dies aus Gründen der Übersichtlichkeit durch ein schwarzes Quadrat in der Zelle angedeutet. Ist das entsprechende Kontrollsignal CTRL_I auf '1' gesetzt, wird das Partialprodukt in der Basiszelle negiert, um der Rechenvorschrift in Gl. (2) bzw. Gl. (3) zu genügen. Die linke untere Basiszelle benötigt diese Negation jedoch nur, wenn sich das Multiplizierer-Element in Position 7 der Zusammenschaltung befindet. Dort muß das Partialprodukt regulär und ohne Invertierung berechnet werden, deshalb kombiniert ein weiteres XOR-Gatter die beiden Signale CTRL_I.1 und CTRL_I.2. Um das korrekte Ergebnis bei der Multiplikation zu erhalten, muß gemäß Hatamian and Cash (1986) und Bermak et al. (1997) ein Korrekturterm addiert werden, der den beiden +1 Anteilen in (2) und (3) entspricht, da eine K2-Zahl durch Umkehren aller Bits und Addieren einer 1 negiert wird. Bei einer Ergebniswortbreite von n_1+n_2 hat dieser Korrekturterm **X** eine '1' an den Stellen $x(n_1+n_2-1)$, $x(n_1-1)$ und $x(n_2-1)$. In der quadratischen Struktur in Abb. 7, die den Sonderfall $n_1=n_2=n$ zeigt, wurde dies auf zwei gesetzte Bits an den Positionen $x(2n-1)$ und $x(n)$ zurückgeführt (Pfänder et al., 2004).

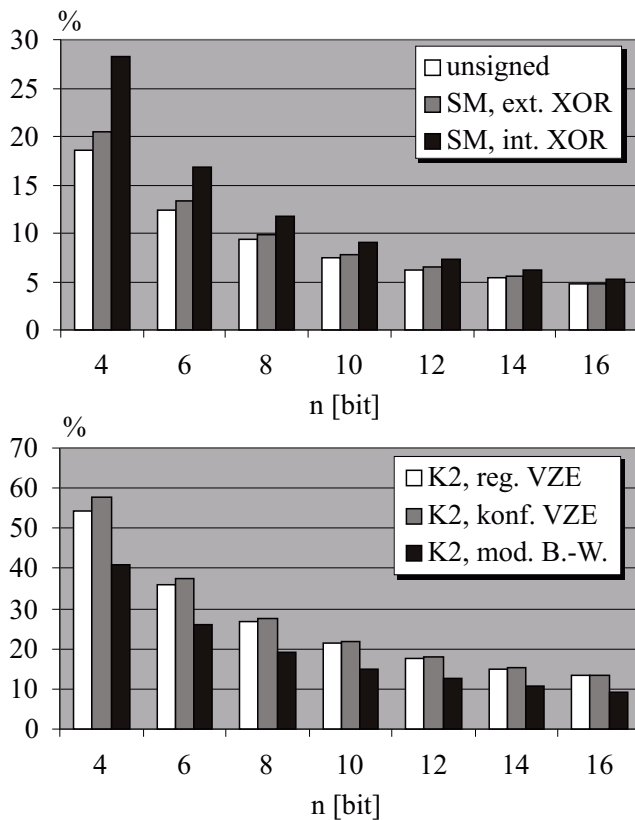


Abbildung 8. Mehraufwand für Konfigurierbarkeit – prozentualer Vergleich nach Pfänder et al. (2004).

2.3 Aufwandsvergleich

Der prozentuale Mehraufwand der verschiedenen konfigurierbaren Multiplizierer-Elemente in Relation zur starren Grundstruktur ohne Schnittstellen wird anhand der Zahl der benötigten Transistoren abgeschätzt Pfänder et al. (2004). Eine grafische Gegenüberstellung ist in Abb. 8 gegeben.

Wenn in der zu implementierenden Anwendung die Multiplikationen häufig mit vorzeichenlosen Zahlen erfolgen und nur selten der Fall einer vorzeichenbehafteten Berechnung auftritt, erscheint die Verwendung eines Signed-Magnitude-Multiplizierers vorteilhaft. Hierbei steht zur Auswahl, die Vorzeichenbehandlung mit einem externen XOR-Gatter oder durch dessen Integration ins Multiplizierer-Element zu realisieren. Bei geringen Basiswortbreiten n von 4 bis 8 bit ist der prozentuale Mehraufwand jedoch sehr groß und kann mit einem internen XOR sogar über 25% betragen. Je größer die Basiswortbreite wird, desto geringer wird der Einfluß der Gatter und Multiplexer, die für die Konfigurierbarkeit und Zusammenschaltbarkeit benötigt werden, auf die Gesamtzahl der Transistoren; z.B. nur noch knapp 5% bei $n=16$, vgl. Abb. 8 (oben).

Bei der Zahldarstellung im Zweierkomplement spiegeln sich die mathematisch bedingte zusätzliche Basiszellenspalte bzw. die modifizierten Basiszellen in der alternativen Variante noch deutlicher im prozentualen Mehraufwand wider,

vgl. Abb. 8 (unten). Während aber bei größer werdendem n die Unterschiede zwischen regulärer und konfigurierbarer Vorzeichenerweiterung nahezu verschwinden, hat die alternative Baugh-Wooley-Variante stets einen deutlich erkennbaren Vorsprung und könnte z.B. bei $n=8$ mit fast 10% weniger Transistoren und damit flächensparender realisiert werden als mit der konfigurierbaren Vorzeichenerweiterung.

3 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Konzept für den Aufbau einer zur Laufzeit rekonfigurierbaren Multiplizierer-Matrix vorgestellt, dessen Elemente jeweils erweiterte und mit Kontrollsignalen dynamisch steuerbare Parallel-Parallel-Multiplizierer sind. Diese Elemente sind selbst separate und voll funktionsfähige Multiplizierer und weisen Schnittstellen auf, die mit Hilfe von Multiplexern realisiert wurden. Mehrere Varianten für vorzeichenlose und vorzeichenbehaftete Zahlen wurden diskutiert. Wenn als Zahldarstellung das Zweierkomplement zum Einsatz kommt, bietet sich für die Multiplizierer-Elemente die modifizierte Struktur nach Baugh-Wooley an, die im Vergleich zur herkömmlichen Vorzeichenerweiterung um bis zu 12% weniger Transistoren benötigt und damit eine höhere Flächeneffizienz bietet.

Ein mögliches Einsatzgebiet solcher zusammenschaltbarer Multiplizierer ist, wie eingangs erwähnt, das Einbetten in Form von fest geschalteten Blöcken innerhalb von FPGA-Strukturen, um deren Logikressourcen bei der Implementierung von Multiplikationen zu entlasten. Die Elemente können dann sowohl im separaten Betrieb als auch als dynamisch verbundener übergeordneter Multiplizierer für viele iterative Algorithmen mit variabler Genauigkeit nützlich sein.

Gegenstand von aktuellen und zukünftigen Überlegungen ist die Gegenüberstellung von verschiedenen Vorgehensweisen bei der Verdrahtung. Eine Möglichkeit ist die Nutzung von FPGA-typischen Routing-Ressourcen, wobei das Multiplizierer-Element durch Hinzunahme z.B. von Schaltmatrizen zur Anbindung an die Infrastruktur bereits als spezialisiertes FPGA-Tile betrachtet werden kann. Bei der Platzierung der Elemente soll die geforderte Lokalität des Carry-Pfades unbedingt berücksichtigt werden, da ansonsten Timing-Probleme auftreten können.

Literatur

- Baugh, C. R. und Wooley, B. A.: A Two's Complement Parallel Array Multiplication Algorithm, IEEE Trans. Computers, C-22, 1045–1047, 1973.
- Bermak, A., Martinez, D., und Noullet, J.-L.: High-Density 16/8/4-bit Configurable Multiplier, IEE Proc. Circuits Devices Systems, 144, 272–276, 1997.
- Bondalapati, K. und Prasanna, V. K.: Reconfigurable Computing Systems, Proc. of the IEEE, 90, 1201–1217, 2002.
- Hatamian, M. und Cash, G. L.: A 70-MHz 7-bit \times 8-bit Parallel Pipelined Multiplier in 2.5- μ m CMOS, IEEE J. Solid-State Circuits, SC-21, 505–513, 1986.

- Haynes, S. D., Ferrari, A. B., und Cheung, P. Y. K.: Flexible Reconfigurable Multiplier Blocks suitable for enhancing the Architecture of FPGAs, Proceedings of the IEEE 1999 Custom Integrated Circuits Conference, San Diego, CA, 191–194, 1999.
- Hwang, K.: Computer Arithmetic – Principles, Architecture, and Design, John Wiley & Sons, New York, 1979.
- Oklobdzija, V. G.: Design of High-Performance Microprocessor Circuits, chap. 10 – High-Speed VLSI Arithmetic Units: Adders and Multipliers, John Wiley & Sons/IEEE Press, New York, 2000.
- Parhami, B.: Computer Arithmetic: Algorithms and Hardware Designs, Oxford University Press, New York, 2000.
- Pfänder, O. A., Hacker, R., und Pfeiderer, H.-J.: A Multiplexer-based Concept for Reconfigurable Multiplier Arrays, Proceedings of the FPL 2004, Antwerp, Belgium, 938–942, 2004.
- Xilinx: Virtex™-II Platform FPGAs and Product Specification, Xilinx Document DS031 (v3.3), 2004.