

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

**Vom Promotionsausschuss der  
Technischen Universität Hamburg-Harburg  
zur Erlangung des akademischen Grades  
Doktor-Ingenieur (Dr.-Ing.)**

**genehmigte Dissertation**

von  
Malte Baesler  
aus Hamburg

2012

1. Gutachter: Prof. Dr.-Ing. Sven-Ole Voigt
2. Gutachter: Prof. Dr.-Ing. Wolfgang Krautschneider

Tag der mündlichen Prüfung: 04.04.2012

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

Copyright Shaker Verlag 2012

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

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](http://www.shaker.de) • e-mail: [info@shaker.de](mailto:info@shaker.de)

# Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Institut für Zuverlässiges Rechnen der Technischen Universität Hamburg-Harburg. An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt und beraten haben.

Mein größter Dank gilt dabei meinem Doktorvater Prof. Dr. Thomas Teufel für die Betreuung und Unterstützung, ohne die diese Arbeit gar nicht erst zustande gekommen wäre. Leider verstarb Prof. Teufel im September 2011, sodass er den Abschluss meiner Promotion nicht mehr persönlich miterleben konnte.

In besonderer Weise bedanke ich mich bei daher bei Herrn Prof. Dr.-Ing. Sven-Ole Voigt für die Übernahme des Erstgutachtens, für die Unterstützung im letzten halben Jahr meiner Promotion sowie für die vielen fachlichen Diskussionen. Darüber hinaus danke ich dem Zweitgutachter Herrn Prof. Dr.-Ing. Wolfgang Krautschneider und dem Vorsitzenden des Prüfungsausschusses Herrn Prof. Dr. Wolfgang Bauhofer.

Ferner bedanke ich mich bei allen Kollegen und ehemaligen Kollegen für die angenehme Arbeitsumgebung, für die stete Unterstützung bei Fragen und Problemen, für die fachlichen Diskussionen und insbesondere für die vielen nicht-fachlichen Konversationen.

Weiterhin danke ich Dennis Ambrahsat und Marcel Fagin, die im Rahmen ihrer Studienarbeiten hilfreiche Beiträge zu meiner Arbeit geliefert haben.

Abschließend gilt mein ganz persönlicher Dank meiner Familie und meinen Freunden für deren Unterstützung und Verständnis.



# Contents

<b>Introduction</b>	<b>1</b>
<b>1. Scientific Computing</b>	<b>5</b>
1.1. Floating-Point Arithmetic . . . . .	5
1.2. Interval Arithmetic . . . . .	11
<b>2. Decimal Floating-Point Arithmetic</b>	<b>17</b>
2.1. Motivation for Decimal Floating-Point Arithmetic . . . . .	17
2.2. IEEE 754-2008 Decimal Floating-Point Arithmetic . . . . .	20
2.2.1. Decimal Encodings . . . . .	20
2.2.2. Rounding . . . . .	22
2.2.3. Arithmetic Operations . . . . .	23
2.2.4. Operations with NaN and Exception Handling . . . . .	24
2.3. Decimal Floating-Point for Scientific Computing . . . . .	25
<b>3. FPGA Technology</b>	<b>27</b>
3.1. Xilinx Virtex-5 FPGA . . . . .	27
3.2. Synthesis and Implementation . . . . .	32
<b>4. Decimal Addition and Subtraction</b>	<b>35</b>
4.1. Decimal Fixed-Point Adder . . . . .	35
4.1.1. Carry-Save Adder . . . . .	36
4.1.2. Carry-Propagate Adder . . . . .	39
4.2. Decimal Fixed-Point Subtractor . . . . .	41
4.3. Decimal Floating-Point Adder/Subtractor . . . . .	42
4.4. Implementation Results . . . . .	52
<b>5. Decimal Multiplication</b>	<b>55</b>
5.1. Decimal Fixed-Point Multiplier . . . . .	55
5.1.1. Multi-operand Decimal Adder Trees . . . . .	57
5.1.2. Proposed Parallel Decimal Fixed-Point Multiplier . . . . .	60
5.2. Decimal Floating-Point Multiplier . . . . .	64
5.3. Implementation Results . . . . .	74

<b>6. Decimal Accurate Scalar Product</b>	<b>81</b>
6.1. Fixed-Point Accurate Scalar Product . . . . .	83
6.1.1. Fixed-Point Multiplier . . . . .	83
6.1.2. Fixed-Point Accumulator . . . . .	83
6.1.3. Pipelining . . . . .	96
6.1.4. Working Spaces . . . . .	96
6.2. Floating-Point Accurate Scalar Product . . . . .	96
6.2.1. Rounding Unit . . . . .	97
6.2.2. Exception Handling . . . . .	103
6.3. Implementation Results . . . . .	104
<b>7. Decimal Division</b>	<b>107</b>
7.1. Decimal Fixed-Point Division . . . . .	107
7.1.1. Functional Iteration . . . . .	108
7.1.2. Digit Recurrence . . . . .	111
7.1.3. Proposed Decimal Fixed-Point Divider . . . . .	121
7.2. Decimal Floating-Point Divider . . . . .	123
7.3. Implementation Results . . . . .	128
<b>8. Decimal Floating-Point Co-Processor</b>	<b>133</b>
8.1. Decimal Arithmetic Unit . . . . .	133
8.2. Verification . . . . .	139
8.3. Implementation Results . . . . .	140
<b>9. Summary and Conclusion</b>	<b>143</b>
9.1. Summary of Contributions . . . . .	144
9.2. Future Work . . . . .	145
<b>A. Implementation of Basic Components</b>	<b>149</b>
A.1. Fast Long Or . . . . .	149
A.2. Leading Zeros / Leading Nines Counters . . . . .	150
A.3. Barrel Shifters . . . . .	151
A.3.1. Parallel, LUT-based Barrel Shifter . . . . .	151
A.3.2. Parallel Multiplier-based Barrel Shifter . . . . .	152
A.3.3. Serial Barrel Shifter . . . . .	154
<b>B. Proofs</b>	<b>155</b>
<b>Glossary</b>	<b>167</b>
<b>Bibliography</b>	<b>176</b>



# List of Figures

1.1.	Examples of a non-normalized and normalized floating-point system. . . . .	8
2.1.	Decimal interchange floating-point formats [IEEE08]. . . . .	21
3.1.	Virtex-5 configurable logic block (CLB). . . . .	28
3.2.	Xilinx <sup>®</sup> Virtex-5 slice (simplified) [Xil09b]. . . . .	29
3.3.	Diagonal symmetric interconnect pattern for Virtex-5 [MK09]. . . . .	30
4.1.	Type1 BCD-4221 carry-save adder implementations. . . . .	38
4.2.	Type2 BCD-4221 carry-save adder implementations. . . . .	38
4.3.	An example of BCD-8421 addition with pre- and post-correction. . . . .	39
4.4.	BCD-8421 adder with pre- and post-correction. . . . .	40
4.5.	BCD-8421 adder with direct carry-out implementation. . . . .	41
4.6.	BCD-8421 inversion and combined adder/subtractor. . . . .	42
4.7.	Block diagram of the decimal floating-point adder. . . . .	44
4.8.	Right shift of the operand $c_B^{\text{swap}}$ . . . . .	46
4.9.	Operand placement for addition a) and subtraction b). . . . .	47
4.10.	Fixed-point adder with injection-based rounding. . . . .	48
5.1.	Simplified Virtex-5 slice with delays T1 - T5. . . . .	59
5.2.	Parallel fixed-point multiplier. . . . .	61
5.3.	Multiplicand Multiples Generator (MMGen). . . . .	62
5.4.	Example of a CSAT with 6 input vectors. . . . .	64
5.5.	Block diagram of DFP multiplier. . . . .	65
5.6.	CPA for delayed CPA solution. . . . .	68
5.7.	Example of a <i>decimal64</i> multiplication with <i>RoundTiesToEven</i> mode. . . . .	73
5.8.	Speed of fixed-point multiplier for different pipeline stages. . . . .	75
5.9.	Speed of floating-point multipliers for different pipeline stages. . . . .	79
6.1.	MAC fixed-point accumulator. . . . .	84
6.2.	Block alignment of the cyclic barrel shifter and the long accumulator. . . . .	87
6.3.	Central (4:2) adder for the operands. . . . .	89
6.4.	Central (4:2) adder for the carry cache. . . . .	90

## List of Figures

---

6.5. Touched blocks register. . . . .	91
6.6. Address generator of the TBR. . . . .	92
6.7. Timing diagram for the accurate scalar product during a MAC operation. . . . .	93
6.8. Final carry-propagate adder. . . . .	94
6.9. Interface of the rounding unit. . . . .	98
6.10. Block diagram of the MAC rounding unit. . . . .	99
6.11. Necessary condition for gradual underflow. . . . .	102
7.1. P-D diagram for the <i>type2</i> digit selection function. . . . .	116
7.2. Block diagram of <i>type2</i> digit recurrence. . . . .	117
7.3. P-D diagram for the <i>type3</i> quotient digit's component $z^3$ . . . . .	119
7.4. Block diagram of <i>type3</i> digit recurrence. . . . .	121
7.5. Block diagram of the normalized <i>type1</i> division. . . . .	122
7.6. Block diagram of the normalized <i>type2</i> and <i>type3</i> division. . . . .	122
7.7. Block diagram of the floating-point divider. . . . .	124
8.1. Block diagram of the decimal arithmetic unit. . . . .	135
8.2. Computation of the selection signals. . . . .	138
A.1. Fast long boolean OR. . . . .	149
A.2. Leading zeros and leading nines counter for 16 digits. . . . .	150
A.3. 8-bit barrel shifter implemented of (2:1) multiplexers. . . . .	151
A.4. 32-bit barrel shifter implemented of LUTs and FxMUXs. . . . .	153
A.5. 48-bit barrel shifter implemented of $16 \times 16$ multipliers. . . . .	154

# List of Tables

1.1. Mathematical spaces and their finite representation on computers. . . . .	6
2.1. Decimal interchange format parameters [IEE08]. . . . .	21
2.2. Decoding a 10-bit declet $b_0$ to $b_9$ into three decimal digits [IEE08]. . . . .	22
2.3. Encoding three decimal digits to a 10-bit declet $b[9 : 0]$ [IEE08]. . . . .	22
3.1. Properties of synthesis and implementation. . . . .	33
4.1. Rounding injection for the decimal floating-point adder. . . . .	50
4.2. Post-place & route results for 16 digits fixed-point adders. . . . .	52
4.3. Post-place & route results for the <i>decimal64</i> floating-point adder. . . . .	53
4.4. Post-place & route results for 64 bit binary floating-point adders. . . . .	53
5.1. Estimated delay and area for decimal multi-operand adder trees. . . . .	60
5.2. Decimal floating-point multiplier: round-up computation. . . . .	70
5.3. Post-place & route results for DFixMul with CPA output. . . . .	74
5.4. Performance comparison of 16-digits fixed-point multipliers. . . . .	75
5.5. Post-place & route results for type1 mul. (mul-based shift, delayed CPA). . . . .	76
5.6. Post-place & route results for type2 mul. (mux-based shift, delayed CPA). . . . .	77
5.7. Post-place & route results for type3 mul. (mux-based shift, immed. CPA). . . . .	78
5.8. Area breakdown of the decimal floating-point multiplier. . . . .	80
5.9. Post-place & route results for 64 bit binary floating-point multipliers. . . . .	80
6.1. Generation of the FCPA BCD-8421 sign extension vector. . . . .	95
6.2. Computation of the MAC rounding injection value. . . . .	103
6.3. Post-place & route results for the fixed-point accurate MAC unit. . . . .	104
6.4. Post-place & route results for the floating-point accurate MAC* unit. . . . .	105
6.5. Runtime of the accurate MAC: software (SW) vs. hardware (HW). . . . .	105
7.1. Constants for type3 quotient digit selection function. . . . .	120
7.2. Round-up detection. . . . .	128
7.3. Post place & route results of the normalized decimal fixed-point dividers. . . . .	129
7.4. Performance comparison of fixed-point dividers. . . . .	130
7.5. Post place & route result of the floating-point divider based on type2. . . . .	131

7.6. Post place & route result of a 64 bit binary floating-point divider (Core-Gen). . . . .	131
8.1. Number of configurable pipeline <sup>1</sup> or synchronization <sup>2</sup> registers. . . . .	139
8.2. Post place & route results of the decimal floating-point unit. . . . .	141
8.3. Resource usage of the co-processor unit on a Virtex-5 XC5VLX330T. . . . .	142
9.1. Interval multiplication. . . . .	146
9.2. Interval division. . . . .	146
A.1. Binary barrel shifter implementation. . . . .	152

# List of Algorithms

1.1. Verified solution of systems of linear equations $Ax = b$ . . . . .	16
4.1. Algorithm of the swapping unit. . . . .	46
4.2. Pseudo code for fixed-point adder. . . . .	49
5.1. Pseudo code of the exponent and significand reset unit. . . . .	66
5.2. Overflow/underflow correction algorithm. . . . .	69
5.3. Decimal floating-point multiplier: rounding algorithm. . . . .	72
6.1. Line, block, and column address generation. . . . .	87
6.2. Pseudo code for LACC calculation. . . . .	88
6.3. Pseudo code for MAC Rounding Unit 1/2. . . . .	100
6.4. Pseudo code for MAC Rounding Unit 2/2. . . . .	101
7.1. Pseudo code for restoring type1 digit recurrence division. . . . .	114
7.2. Algorithm for the calculation of the type2 ROM entries. . . . .	116
7.3. Pseudo code for the type2 digit recurrence. . . . .	117
7.4. Pseudo code for the type3 digit recurrence. . . . .	120
7.5. Algorithm of the NaN handling unit. . . . .	125
7.6. On-the-fly conversion with gradual underflow handling. . . . .	126
7.7. Exponent calculation. . . . .	127
8.1. Scoreboarding Algorithm for a single critical component. . . . .	136
8.2. Locking signals generation. . . . .	137