

A Flexible Multi-Processor System-on-a-Chip Architecture for Safety- and Security-Critical Applications

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

von
Daniel Kliem

aus
Stade

2013

1. Gutachter: Prof. Dr.-Ing. Sven-Ole Voigt
2. Gutachter: Prof. Dr. Fritz Mayer-Lindenberg

Tag der mündlichen Prüfung: 18.07.2013

Technische Informatik

Daniel Kliem

**A Flexible
Multi-Processor System-on-a-Chip Architecture
for Safety- and Security-Critical Applications**

Shaker Verlag
Aachen 2013

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., 2013

Copyright Shaker Verlag 2013

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 978-3-8440-2268-1

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

Danksagung

Diese Dissertation entstand während meiner wissenschaftlichen Tätigkeit als Gast im Institut für Zuverlässiges Rechnen der Technischen Universität Hamburg-Harburg. Ich danke all denjenigen, die mir dieses Projekt durch ihre Beratung und Unterstützung ermöglicht haben.

Meinem Doktorvater Herrn Prof. Dr.-Ing. Sven-Ole Voigt danke ich besonders für seine sorgfältige Betreuung und die Übernahme des Erstgutachtens. Die ausführlichen Diskussionen waren eine wertvolle und unerlässliche Stütze. Zudem danke ich meinem Zweitgutachter Herrn Prof. Dr. Fritz Mayer-Lindenberg sowie dem Vorsitzenden des Prüfungsausschusses Herrn Prof. Dr.-Ing. Wolfgang Krautschneider.

Dankend erwähnen möchte ich außerdem den leider zwischenzeitlich verstorbenen Herrn Prof. Dr. Thomas Teufel, der mich in der ersten Phase meines Promotionsvorhabens beraten hat.

Bei den Mitarbeiterinnen und Mitarbeitern des Institutes bedanke ich mich für die herzliche Aufnahme in ihre Mitte und für die angenehme Arbeitsumgebung mit vielen fachlichen und persönlichen Gesprächen und guten Ratschlägen.

Ebenso danke ich den Kolleginnen und Kollegen bei AIRBUS, die mich mit zahlreichen Diskussionen unterstützten und mir ein Umfeld geboten haben, das diese Arbeit ermöglichte.

Den Herren Ivan Steiger, Sebastian Hoop, Christian Sauer, Kai Torben Ohlhus und Stefan Gehrler danke ich für ihre hilfreichen Beiträge im Rahmen ihrer Bachelor-, Studien- bzw. Masterarbeiten.

Nicht zuletzt danke ich herzlich meinen Freunden, meiner Familie und insbesondere meiner Frau, die mich stets motivierten und mir mit Rat und Tat beiseite standen.

Hamburg, September 2013

Daniel Kliem

Contents

1	Introduction	1
2	Domain Segregation	7
2.1	Definition and Motivation	7
2.2	Two Aspects of Domain Segregation	9
2.3	Safety	10
2.3.1	IEC-61508 Safety Levels	11
2.3.2	Avionics Safety Levels by DO-178B	11
2.3.3	Segregation and DO-178B	15
2.4	Security	15
2.5	Comparing Safety and Security Standards	17
3	Current Segregation Solutions	19
3.1	Evaluation Criteria and Application Scenarios	19
3.2	Solution 1: Custom Multi-Core Software Implementations	22
3.3	Solution 2: Partitioning Operating Systems	24
3.4	Solution 3: Virtual Machines	27
3.5	Solution 4: Custom Multi-Core Hardware Implementations	29
3.6	Evaluation of Current Solutions	30
4	Proposed Architecture	33
4.1	Architecture Requirements and Concepts	34
4.2	Related Work	35
4.3	Architecture and Secure Bus Bridges	36
4.4	Designing for Temporal Segregation	38
4.5	A Detailed Implementation Example	39
4.6	Communication Across Segregation Boundaries	41
4.7	Support for Software Re-Use	42
4.8	Assumptions	43

4.9	Address Translation	43
4.10	A Certification Strategy Example	44
5	Prototype Design	49
5.1	GRLIB and AMBA	50
5.2	Xilinx Virtex 6	53
5.3	Behavioral Memory Simulations	54
5.4	Prototype Design	55
6	Bus Performance Monitoring	57
6.1	Bus Observer Unit – AHBOBS	58
6.2	Slave Observer Unit – AHBSLVOBS	62
6.3	Synchronization Unit – APBSYNC	63
6.4	Simulation Ballast Program	64
6.5	Scalability Evaluation	66
7	Verification Methods and Regression Testing	69
7.1	Conventional Approach	69
7.2	In-Situ Testing	70
7.3	Test Program Code Generation	71
8	Basic Secure Bridges	73
8.1	Common Principles	74
8.2	Synchronous Combinatorial Bridge	74
8.2.1	Bridge Internals	74
8.2.2	Wait-State Profile	76
8.2.3	Interruptible Bursts	77
8.2.4	Influence of Backbone Wait-States	78
8.2.5	Implementation Results and Conclusion	80
8.3	Synchronous Cached Bridge	82
8.3.1	Bridge Internals	82
8.3.2	Cache Operation	85
8.3.3	Wait-State Profile	86
8.3.4	Influence of Cache Capacity and Organization	88
8.3.5	Influence of Backbone Data Width	89
8.3.6	Implementation Results	91
9	Asynchronous Cached Secure Bridge	93
9.1	Bridge Building Blocks	94
9.2	Asynchronous Queues	95
9.3	Cache Organization	96

9.4	Cache Memory Layout	97
9.5	Cache Operation	98
9.6	Wait-State Profile	100
9.7	Backbone Access Optimization	101
9.8	Influence of Cache Capacity and Organization	105
9.9	Influence of Pre-Fetching	106
9.10	Influence of Backbone Data Width	107
9.11	Influence of Clock Ratio	108
9.12	Influence of Backbone Wait-States	109
9.13	Timing Performance and Resource Usage	109
9.14	Bridge Type Comparison	111
10	Implementation Aspects of Large Scale FPGA Designs	113
10.1	Clock Domains	114
10.2	Clock and Reset Distribution	115
10.3	Large FPGA Designs	117
11	Evaluation of Secure Bridge Implementations	121
11.1	Implementation Overview	121
11.2	Native Benchmark Suite	123
11.3	Benchmark Suite Results	126
11.3.1	Influence of Pre-Fetching	126
11.3.2	Influence of Cache Capacity and Organization	128
11.3.3	Bridge Type Comparison	130
11.4	High-Level Linux Benchmarks	130
11.5	Linux Benchmark Results	133
11.6	Summary of Implementation Benchmarks	136
12	Further Concepts and Future Work	137
12.1	Pending Evaluations	137
12.2	Further Concepts	138
13	Summary and Conclusion	141
	Appendices	145
A	Survey of Embedded MPSoC Implementations	147
A.1	MPSoC	147
A.2	Programmable SoC Platforms	149

B Working with the Architecture	153
B.1 Regular Power-Up and Initialization Sequence	153
B.2 Debugging of Multiple Systems via Ethernet	155
B.3 Running Linux Systems	157
C Additional Simulation Results	159
D Additional Implementation Results	165
E Cache Controller State Machine of the Asynchronous Bridge	169
F Resource and Timing Evaluation	173
F.1 Resource Usage	173
F.2 Timing	174
G AHB2AHB Bridge by Aeroflex Gaisler	177
G.1 Description	177
G.2 Brief Evaluation Results	178
Acronyms	181
Bibliography	185

List of Figures

1.1	Emergence of IMA	2
1.2	Memory Wall	2
2.1	Certification effort of selected safety and security standards	18
3.1	Application scenario: Mixing safety levels	20
3.2	Application scenario: A network gateway	21
3.3	Segregation Solution 1: Multi-processor SoC	23
3.4	Segregation Solution 2: Software partitioning	25
3.5	Software partition timing	27
3.6	Segregation Solution 3: Virtualization	28
3.7	Segregation Solution 4: Programmable SoC	29
4.1	Proposed architecture: Multi-SoC system	34
4.2	Architecture with physical domain partitioning	36
4.3	Proposed architecture with domain segregation	37
4.4	Segregation concept	38
4.5	Backbone transfer combining	39
4.6	Application example: Heterogeneous triple system	40
4.7	Proposed communication architecture	41
4.8	Comparison of segregation techniques	46
5.1	AMBA AHB multiplexed bus structure	51
5.2	AMBA AHB pipeline principle	53
5.3	Backbone memory	54
5.4	Bus structure and VHDL entities	56
6.1	System evaluation setup	58
6.2	AHB observer unit	61
6.3	AHBOBS write pipeline	61

List of Figures

6.4	APBSYNC unit	63
6.5	Relationship of RET and BIR	67
7.1	Verification toolchain	71
7.2	Test generator mirror principle	72
8.1	Internal structure of synchronous combinatorial bridge	75
8.2	State machine of synchronous combinatorial bridge	75
8.3	Wait-states of combinatorial bridge	76
8.4	Simulation: Scalability of the synchronous combinatorial bridge	79
8.5	Internal structure of synchronous cached bridge	82
8.6	Robust placement of address translator	83
8.7	Cache memories of synchronous bridge	84
8.8	Timing of synchronous cached bridge	86
8.9	Wait-states of synchronous bridge.	87
8.10	Simulation: Cache sizes and synchronous cached bridge	89
8.11	Simulation: Backbone width and synchronous cached bridge	90
9.1	Internal structure of asynchronous cached bridge	94
9.2	Cache memories of asynchronous bridge	97
9.3	Two-way set-associative cache memories	98
9.4	State machine of asynchronous bridge	99
9.5	Wait-states of asynchronous bridge	101
9.6	Burst unit operation	102
9.7	Simulation: Cache sizes and asynchronous cached bridge	104
9.8	Simulation: Pre-fetching and asynchronous cached bridge	106
9.9	Simulation: Backbone width and asynchronous cached bridge	107
9.10	Simulation: Backbone frequency and asynchronous cached bridge	109
9.11	Simulation: Backbone wait-state and asynchronous cached bridge	110
9.12	Simulation: Bridge type comparison	111
10.1	Design clock partitioning	114
10.2	Virtex 6 clock regions	115
10.3	Local reset generator	116
10.4	Reset generator implementation	116
10.5	Floorplan of a 1x1 SoSoC design	118
10.6	Floorplan of a 2x2 SoSoC design	118
10.7	Floorplan of a 4x1 SoSoC design	119
10.8	Floorplan of a 4x1 SoSoC design	119
11.1	Benchmark: Pre-fetching and asynchronous cached bridge	127

11.2 Benchmark: Cache size comparison	128
11.3 Benchmark: Bridge type comparison	129
11.4 Linux benchmark results	135
A.1 MPSoC P5040	148
A.2 MPSoC i.MX6	148
A.3 TI MPSoC OMAP	149
A.4 Xilinx Zynq 7000 SoC FPGA	150
A.5 Xilinx Zynq memory interface	151
A.6 Altera SoC FPGA	151
B.1 System memory layout for power-up	154
B.2 Multi-system Ethernet setup for debug access	156
C.1 Simulation: Wait-states and synchronous cached bridge	160
C.2 Simulation: Cache sizes and synchronous cached bridge	161
C.3 Simulation: Scalability of the asynchronous cached bridge	162
C.4 Simulation: Wait-states and asynchronous cached bridge	163
C.5 Simulation: Cache size and asynchronous cached bridge	164
D.1 Benchmark: Bridge type comparison	166
D.2 Benchmark: Pre-fetching and asynchronous cached bridge	167
D.3 Benchmark: Cache size and bridge types	168
E.1 State machine of asynchronous bridge	170
F.1 Harness for resource evaluation	174
F.2 Harness for timing evaluation	174
G.1 MPSoC Quad-core LEON4	178
G.2 Wait-states of AHB2AHB bridge	179
G.3 Simulation: Scalability of the Gaisler AHB2AHB bridge	179
G.4 Simulation: Backbone width and sync. AHB2AHB bridge	180

List of Tables

1.1	Clock Speed Discrepancy (Vendor Data)	3
2.1	Design Assurance Levels of DO-178B	14
2.2	Common Criteria Evaluation Assurance Levels	17
3.1	Comparison of Current Segregation Implementations	32
6.1	AHB Control Signal to Counter Mappings	60
8.1	Synchronous Combinatorial Bridge Implementation Details	80
8.2	Transfer Statistics of Synchronous Combinatorial Bridge	81
8.3	Cache Address Fields of the Synchronous Cached Bridge	84
8.4	Transfer Statistics of Synchronous Cached Bridge	88
8.5	Synchronous Cached Bridge Implementation Details	91
9.1	Cache Address Fields of the Asynchronous Cached Bridge	96
9.2	Transfer Statistics of Asynchronous Cached Bridge	103
9.3	Asynchronous Cached Bridge Implementation Details	110
11.1	Implemented Designs on Virtex 6 LX240T	122
11.2	Transfer Statistics of Benchmark Suite	126
11.3	Transfer Statistics of Linux Benchmarks	132