

A Reconfigurable Prototyping System for Multiple-Input Multiple-Output Communications

by

John William Dalton
BSc, B.E. (Hons. I)

*A thesis submitted in total fulfilment
of the requirements of the degree of
Master of Engineering*

School of Electrical Engineering and Computer Science
The University of Newcastle
Australia

November, 2009



THE UNIVERSITY OF
NEWCASTLE
AUSTRALIA

© Copyright by John William Dalton 2009

In addition to the University of Newcastle's deposit license, this thesis is licensed under the terms of the Creative Commons Attribution-ShareAlike 2.5 Australia License. Further, the appendices can be redistributed and/or modified under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. The full text of each license is in the end matter of this thesis. Source code can be downloaded from <http://john.daltons.info/masters/> and the author can be contacted on john.dalton@daltons.info.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying subject to the provisions of the Copyright Act 1968.

John William Dalton

For Claire and the children

Acknowledgements

Many thanks are due to my supervisor, Associate Professor Steve Weller. His voice of experience has been indispensable in completing this thesis. Thank you Steve. Thank you also to my co-supervisor, Professor Brett Ninness.

My work was partially funded by an Australian Postgraduate Award. Funding to attend conferences and workshops was provided by the ARC Communications Research Network (ACoRN). Early funds were also provided by ARC Linkage Project Grant LP0211210.

Thanks are due to my father, Tim Dalton, for his careful proofreading of this thesis. I am also very appreciative of Steve's many proofreads and suggestions. I would insert a responsibility statement here, but the task exceeds my abilities [1].

Finally, a loving thank you to Claire and my family. It is you who have sustained me through the academic process. Thank you for your patience, support and love.

Contents

Acknowledgements	ix
Abstract	xvii
1 Introduction	1
1.1 MIMO Background	1
1.2 Programmable Logic for Prototyping	3
1.3 Motivation	4
1.4 Organisation of the Thesis	6
1.5 Original Contributions of the Thesis	7
2 Hardware	9
2.1 System Architecture	10
2.1.1 Modularisation	13
2.2 PCI Mainboard	18
2.2.1 Programmable Logic	21
2.2.2 Power Supply	24
2.2.3 External Data: PCI Bridge	26
2.2.4 Clocking	28
2.2.5 Inter-Board Signalling	32
2.2.6 Printed Circuit Board Layout	33
2.3 2.4 GHz Radio Transceiver	39
2.3.1 Power Supplies	42
2.3.2 RF system	44
2.3.3 Analogue-to-Digital Conversion	48
2.3.4 Digital-to-Analogue Conversion	55

2.3.5	Low-Speed Digital-to-Analogue Converter	61
2.3.6	Low-Speed Analogue-to-Digital Converter	64
2.3.7	Clocking	65
2.3.8	PCB Layout	68
2.4	Antenna	70
3	Firmware	75
3.1	VHDL Programming	76
3.1.1	Device Independence	78
3.1.2	Design Flow	80
3.2	Input/Output Layer	82
3.3	Abstraction Layer	83
3.3.1	Transceiver Components	84
3.3.2	Transceiver	95
3.3.3	PCI Interface	100
3.4	Test Benches	102
3.5	Core Logic	106
3.5.1	Configuration Test	110
3.5.2	PCI Bus Test	110
3.5.3	Network Analyser	110
4	Software	119
4.1	Driver Software	119
4.1.1	Application Programming Interface	121
4.1.2	Internal Structure	123
4.1.3	Building the Driver	135
4.1.4	Installing the Driver	137
4.2	Abstraction Library	137
4.2.1	Testbed Card Class	138
4.2.2	Transceiver Class	140
4.2.3	Network Analyser Class	143
4.2.4	Using the Abstraction Library	145
4.3	Application Software	145
4.3.1	PCI Bus Test	146
4.3.2	Command Line Shell	146

4.4	A Multiple Antenna Communications System	147
4.4.1	The Channel Class	148
4.4.2	Modulation and Coding	160
4.4.3	Generating Bit Error Rate Performance Curves	165
4.4.4	Top-Level Scripts	166
5	Results	167
5.1	Transmitting and Receiving a Single Tone	168
5.2	Dual Tones on a 2×2 Channel	171
5.3	Simulated BER Performance Curves	171
5.4	Estimation Loss	175
5.5	Bit Error Rate as a Function of Operating Point	176
5.6	Measured BER Performance	178
6	Conclusion	181
6.1	Future Work	183
A	Mainboard Schematics	185
B	Transceiver Schematics	203
C	Antenna Schematic	213
D	Printed Circuit Board Layout for the Mainboard	215
D.1	Source Code for Routing the FPGA	222
E	Printed Circuit Board Layout for the Transceiver	227
F	Printed Circuit Board Layout for the Antenna	235
G	VHDL I/O Layer Source Code	239
G.1	fpga_toplevel.vhdl	239
G.2	iob_primitives.vhdl	269
H	VHDL Abstraction Layer Source Code	277
H.1	testbed_ring_of_pads.vhdl	277
H.2	transceiver_2_4_ghz_registered.vhdl	288
H.3	transceiver_2_4_ghz.vhdl	294

H.4	max5189.vhdl	297
H.5	ths0842.vhdl	298
H.6	ad5314.vhdl	298
H.7	max1243.vhdl	300
H.8	max2822.vhdl	302
H.9	pci9052.vhdl	304
I	VHDL Core Logic Source Code	307
I.1	network_analyser_memory.vhdl	307
I.2	dp_mux_block_ram.vhdl	311
I.3	synchronous_block_ram.vhdl	312
I.4	sp_block_ram.vhdl	314
J	VHDL Test Bench Source Code	317
J.1	network_analyser_testbench.vhdl	317
J.2	fpga_device_wrapper_model.vhdl	334
J.3	transceiver_2_4_GHz_model.vhdl	342
J.4	max5189_model.vhdl	345
J.5	ths0842_model.vhdl	347
J.6	ad5314_model.vhdl	349
J.7	max1243_model.vhdl	352
J.8	max2822_model.vhdl	354
J.9	pci9052_model.vhdl	357
K	Driver Software Source Code	365
K.1	testbed.c	365
K.2	Makefile	372
L	Abstraction Library Source Code	373
L.1	network_analyser_configure_transceivers.m	373
L.2	network_analyser_constructor.m	373
L.3	network_analyser_read.m	374
L.4	network_analyser_read_buffer.m	374
L.5	network_analyser_run.m	375
L.6	network_analyser_trigger.m	375
L.7	network_analyser_write.m	376

L.8	network_analyser_write_buffer.m	376
L.9	testbed_card_constructor.m	377
L.10	testbed_card_read_attribute.m	377
L.11	testbed_card_read_register.m	377
L.12	testbed_card_write_attribute.m	378
L.13	testbed_card_write_register.m	378
L.14	testbed_transceiver_base_address.m	379
L.15	transceiver_channel_register.m	379
L.16	transceiver_configure.m	379
L.17	transceiver_constructor.m	380
L.18	transceiver_control_register.m	380
L.19	transceiver_enable_register.m	381
L.20	transceiver_read_tx_power.m	382
L.21	transceiver_read_tx_power_dbm.m	382
L.22	transceiver_receiver_register.m	382
L.23	transceiver_rx_gain_dac_register.m	383
L.24	transceiver_set_rx_gain_dB.m	383
L.25	transceiver_set_to_receive.m	384
L.26	transceiver_set_to_transmit.m	384
L.27	transceiver_set_tx_gain_dB.m	384
L.28	transceiver_set_tx_power.m	385
L.29	transceiver_synthesiser_register.m	385
L.30	transceiver_transmitter_register.m	385
L.31	transceiver_tx_gain_dac_register.m	386

M Application Software Source Code 387

M.1	alamouti_link.m	387
M.2	channel_constructor.m	388
M.3	channel_insert_silence.m	389
M.4	channel_insert_training_sequence.m	389
M.5	channel_make_consistent.m	389
M.6	channel_model.m	390
M.7	channel_randomise.m	391
M.8	channel_remove_training_sequence.m	392

M.9 channel_testbed.m	393
M.10 channel_tx_rx.m	393
M.11 delay.m	394
M.12 estimate_snr.m	395
M.13 ml_decoder.m	395
M.14 modulate.m	395
M.15 rand_rayleigh.m	396
M.16 single_antenna_link.m	396
M.17 waterfall_curve.m	397
M.18 waterfall_histogram.m	398
M.19 waterfall_plot.m	399
M.20 alamouti_1998_figure_4.m	399
M.21 testbed_1x1.m	400
M.22 testbed_2x2.m	401
Copyright Licenses	403
GNU General Public License	403
Creative Commons BY-SA 2.5 Australia License	410
List of Tables	415
List of Figures	421
Bibliography	423

Abstract

This thesis demonstrates the process of building a system to test multiple-input multiple-output (MIMO) communications over-the-air. It covers the entire process, from concept to design and construction, culminating in transmitting space-time coded data packets and producing bit error rate (BER) performance curves.

A flexible modular architecture is designed, able to test current MIMO systems and to be upgraded as the field develops. Printed circuit boards for a field-programmable gate array (FPGA) based mainboard, 2.4 GHz transceivers and antennas are then designed, embodying the aforementioned architecture. The mainboard uses a Xilinx XC2S600E FPGA, with $\sim 600,000$ logic gates. Hardware is assembled and tested, forming a foundation for further layers of firmware and software. An abstraction layer, with associated test benches, is written in a hardware description language (VHDL), allowing the core logic of the FPGA to be written and simulated in a device-independent manner. Further VHDL is written and the testbed configured to transmit and receive bursts of data. A device driver is implemented, and abstract data types are layered on top of the driver, enabling high-level control of the testbed. Single antenna and MIMO data links are implemented using 1×1 binary phase-shift keying (BPSK) and 2×2 Alamouti encoded BPSK modulation respectively. Finally, data packets are transmitted and measured BER performance curves constructed.

Channel estimation is proved to work on a 2×2 MIMO channel over-the-air, the introduced loss of E_b/N_0 shown to be approximately 0.5 dB compared to perfect channel information. The analogue limitations of the hardware are investigated and bit error rate performance measured as a function of operating point. Finally single antenna communications and a 2×2 Alamouti MIMO scheme are compared over-the-air, the Alamouti scheme delivering a 3 dB improvement in E_b/N_0 performance. Satisfyingly the MIMO scheme also exceeds the best case theoretical performance bound of the single antenna case by a margin of 2 dB in E_b/N_0 .