

Contents

1	Introduction	1
1.1	Overview	1
1.2	Objectives	6
1.3	Outline	6
	References	7
2	3GPP Long Term Evolution	9
2.1	Introduction	9
2.1.1	Evolution and Environment of 3GPP Telecommunication Systems	9
2.1.2	Terminology and Requirements of LTE	10
2.1.3	Scope and Organization of the LTE Study	12
2.2	From IP Packets to Air Transmission	14
2.2.1	Network Architecture	14
2.2.2	LTE Radio Link Protocol Layers	15
2.2.3	Data Blocks Segmentation and Concatenation	17
2.2.4	MAC Layer Scheduler	18
2.3	Overview of LTE Physical Layer Technologies	18
2.3.1	Signal Air transmission and LTE	18
2.3.2	Selective Channel Equalization	21
2.3.3	eNodeB Physical Layer Data Processing	22
2.3.4	Multicarrier Broadband Technologies and Resources	22
2.3.5	LTE Modulation and Coding Scheme	28
2.3.6	Multiple Antennas	31
2.4	LTE Uplink Features	33
2.4.1	Single Carrier-Frequency Division Multiplexing	33
2.4.2	Uplink Physical Channels	35
2.4.3	Uplink Reference Signals	36
2.4.4	Uplink Multiple Antenna Techniques	38
2.4.5	Random Access Procedure	39

2.5	LTE Downlink Features	42
2.5.1	Orthogonal Frequency Division Multiplexing Access . . .	42
2.5.2	Downlink Physical Channels	44
2.5.3	Downlink Reference Signals.	45
2.5.4	Downlink Multiple Antenna Techniques	46
2.5.5	UE Synchronization	49
	References	50
3	Dataflow Model of Computation	53
3.1	Introduction	53
3.1.1	Model of Computation Overview	53
3.1.2	Dataflow Model of Computation Overview	55
3.2	Synchronous Data Flow	59
3.2.1	SDF Schedulability	59
3.2.2	Single Rate SDF.	62
3.2.3	Conversion to a Directed Acyclic Graph	62
3.3	Interface-Based Synchronous Data Flow.	63
3.3.1	Special Nodes.	63
3.3.2	Hierarchy Deadlock-Freeness	64
3.3.3	Hierarchy Scheduling	66
3.3.4	Hierarchy Behavior	67
3.3.5	Hierarchy Improvements	68
3.4	Cyclo Static Data Flow	68
3.4.1	CSDF Schedulability	69
3.5	Dataflow Hierarchical Extensions	69
3.5.1	Parameterized Dataflow Modeling.	70
3.5.2	Interface-Based Hierarchical Dataflow.	72
	References	75
4	Rapid Prototyping and Programming Multi-Core Architectures	77
4.1	Introduction	77
4.1.1	The Middle-Grain Parallelism Level	77
4.2	Modeling Multi-Core Heterogeneous Architectures	79
4.2.1	Understanding Multi-Core Heterogeneous Real-Time Embedded DSP MPSoC.	79
4.2.2	Literature on Architecture Modeling	80
4.3	Multi-Core Programming	82
4.3.1	Middle-Grain Parallelization Techniques	82
4.3.2	PREESM Among Multi-Core Programming Tools	84
4.4	Multi-Core Scheduling.	92
4.4.1	Multi-Core Scheduling Strategies	92
4.4.2	Scheduling an Application Under Constraints.	93
4.4.3	Existing Work on Scheduling Heuristics	95

4.5	Generating Multi-Core Executable Code	98
4.5.1	Static Multi-Core Code Execution.	99
4.5.2	Managing Application Variations	99
4.6	Methods for the LTE Study	100
	References	100
5	A System-Level Architecture Model	103
5.1	Introduction	103
5.1.1	Target Architectures	103
5.1.2	Building a New Architecture Model	107
5.2	The System-Level Architecture Model	107
5.2.1	The S-LAM Operators	107
5.2.2	Connecting Operators in S-LAM.	108
5.2.3	Examples of S-LAM Descriptions.	109
5.2.4	The Route Model	112
5.3	Transforming the S-LAM Model into the Route Model	113
5.3.1	Overview of the Transformation	113
5.3.2	Generating a Route Step	114
5.3.3	Generating Direct Routes from the Graph Model	114
5.3.4	Generating the Complete Routing Table	115
5.4	Simulating a Deployment Using the Route Model	116
5.4.1	The Message Passing Route Step Simulation with Contention Nodes	116
5.4.2	The Message Passing Route Step Simulation Without Contention Nodes	117
5.4.3	The DMA Route Step Simulation	117
5.4.4	The Shared Memory Route Step Simulation.	117
5.5	Role of S-LAM in the Rapid Prototyping Process	117
5.5.1	Storing an S-LAM Graph.	118
5.5.2	Hierarchical S-LAM Descriptions	120
	References	120
6	Enhanced Rapid Prototyping	123
6.1	Introduction	123
6.1.1	The Multi-Core DSP Programming Constraints	123
6.1.2	Objectives of a Multi-Core Scheduler	125
6.2	A Flexible Rapid Prototyping Process	125
6.2.1	Algorithm Transformations While Rapid Prototyping	126
6.2.2	Scenarios: Separating Algorithm and Architecture.	128
6.2.3	Workflows: Flows of Model Transformations.	130
6.3	The Structure of the Scalable Multi-Core Scheduler	133
6.3.1	The Problem of Scheduling a DAG on an S-LAM Architecture	134
6.3.2	Separating Heuristics from Benchmarks.	134

6.3.3	Proposed ABC Sub-Modules	136
6.3.4	Proposed Actor Assignment Heuristics	137
6.4	Advanced Features in Architecture Benchmark Computers	138
6.4.1	The Route Model in the AAM Process	138
6.4.2	The Infinite Homogeneous ABC	139
6.4.3	Minimizing Latency and Balancing Loads	139
6.5	Scheduling Heuristics in the Framework	143
6.5.1	Assignment Heuristics	144
6.5.2	Ordering Heuristics	146
6.6	Quality Assessment of a Multi-Core Schedule	146
6.6.1	Limits in Algorithm Middle-Grain Parallelism	147
6.6.2	Upper Bound of the Algorithm Speedup	148
6.6.3	Lowest Acceptable Speedup Evaluation	149
6.6.4	Applying Scheduling Quality Assessment to Heterogeneous Target Architectures	150
	References	151
7	Dataflow LTE Models	153
7.1	Introduction	153
7.1.1	Elements of the Rapid Prototyping Framework	153
7.1.2	SDF4J: A Java Library for Algorithm Graph Transformations	154
7.1.3	Graphiti: A Generic Graph Editor for Editing Architectures, Algorithms and Workflows	154
7.1.4	PREESM: A Complete Framework for Hardware and Software Codesign	156
7.2	Proposed LTE Models	156
7.2.1	Fixed and Variable eNodeB Parameters	156
7.2.2	A LTE eNodeB Use Case	157
7.2.3	The Different Parts of the LTE Physical Layer Model	159
7.3	Prototyping RACH Preamble Detection	159
7.3.1	Architecture Exploration	162
7.4	Downlink Prototyping Model	163
7.5	Uplink Prototyping Model	166
7.5.1	PUCCH Decoding	166
7.5.2	PUSCH decoding	167
	References	170
8	Generating Code from LTE Models	173
8.1	Introduction	173
8.1.1	Execution Schemes	173
8.1.2	Managing LTE Specificities	175
8.2	Static Code Generation for the RACH-PD Algorithm	176
8.2.1	Static Code Generation in the PREESM Tool	176
8.2.2	Method Employed for the RACH-PD Implementation	179

- 8.3 Adaptive Scheduling of the PUSCH 182
 - 8.3.1 Static and Dynamic Parts of LTE PUSCH Decoding. 182
 - 8.3.2 Parameterized Descriptions of the PUSCH. 183
 - 8.3.3 A Simplified Model of Target Architectures. 185
 - 8.3.4 Adaptive Multi-Core Scheduling of the LTE PUSCH 186
 - 8.3.5 Implementation and Experimental Results 191
- 8.4 PDSCH Model for Adaptive Scheduling. 194
- 8.5 Combination of Three Actor-Level LTE Dataflow Graphs 195
- References 196
- 9 Conclusion. 197**
- Glossary 199**
- Index 207**

Physical Layer Multi-Core Prototyping
A Dataflow-Based Approach for LTE eNodeB

Pelcat, M.; Aridhi, S.; Piat, J.; Nezan, J.-F.

2013, XVI, 212 p., Hardcover

ISBN: 978-1-4471-4209-6