

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Evolution and Trends in Embedded Systems.	2
1.1.1	Embedded Hardware	3
1.1.2	Embedded Software	5
1.1.3	Design Gaps.	6
1.2	Tool Flow Requirements: Closing the SW Gap.	7
1.3	MAPS: A Framework for MPSoC Programming	10
1.4	Programming Tool Flows	11
1.5	Synopsis and Outline	13
<b>2</b>	<b>Background and Problem Definition</b>	<b>15</b>
2.1	Motivational Example	15
2.2	Preliminaries.	17
2.2.1	Mapping and Scheduling	17
2.2.2	Sequential Code Performance Estimation.	19
2.2.3	Notation.	21
2.3	Multi-Application Problem	21
2.3.1	Definitions	21
2.3.2	Problem Statement	28
2.4	Sequential Code Problem	28
2.4.1	Compilers and Parallelism Extraction	29
2.4.2	Problem Statement	38
2.5	Parallel Code Problem	39
2.5.1	Process Networks and Dataflow Models	39
2.5.2	The C for Process Networks Language	44
2.5.3	KPN Mapping and Scheduling	44
2.5.4	Problem Statement	46
2.6	Software Defined Radio Problem	47
2.6.1	The Nucleus Project	48
2.6.2	Extensions to the Parallel Problem	49
2.6.3	Problem Statement	52
2.7	Synopsis.	52

<b>3</b>	<b>Related Work</b>	53
3.1	Runtime Management	53
3.1.1	Survey of Runtime Managers	53
3.1.2	This Book in Perspective	55
3.2	Parallelism Extraction from Sequential Code	56
3.2.1	General-Purpose Domain	56
3.2.2	Embedded Domain	57
3.2.3	Commercial Solutions	59
3.2.4	This Book in Perspective	59
3.3	Synthesis of Parallel Specifications	60
3.3.1	General-Purpose Parallel Programming Models	60
3.3.2	Embedded Parallel Programming	61
3.3.3	This Book in Perspective	65
3.4	Software Defined Radio	66
3.4.1	SDR Approaches	66
3.4.2	This Book in Perspective	68
3.5	Multiple Applications	68
3.5.1	Handling Multiple Applications	69
3.5.2	This Book in Perspective	71
<b>4</b>	<b>MPSoC Runtime Management</b>	73
4.1	OSIP Solution	73
4.1.1	Mapping and Scheduling Approach	74
4.1.2	OSIP Architecture	76
4.1.3	Implementation Results	78
4.1.4	OSIP Firmware	79
4.2	Platform Integration	80
4.2.1	Hardware Integration	80
4.2.2	Software Integration	80
4.2.3	System Level Modeling	81
4.3	Benchmarking	84
4.3.1	Experimental Setup	84
4.3.2	Synthetic Benchmarking	85
4.3.3	H.264 Application	86
4.4	Synopsis	87
<b>5</b>	<b>Sequential Code Flow</b>	89
5.1	Tool Flow Overview	89
5.1.1	Clang and LLVM	90
5.1.2	Tool Flow Components	91
5.2	Application Analysis	93
5.2.1	Application Tracing	93
5.2.2	Building the Application Model	95
5.2.3	Sequential Performance Estimation Revisited	97
5.2.4	Analysis Results	98

5.3	Partitioning and Parallelism Extraction. . . . .	99
5.3.1	Graph Clustering. . . . .	99
5.3.2	Discovering Parallelism Patterns. . . . .	100
5.3.3	Global Analysis . . . . .	105
5.4	Backend . . . . .	108
5.4.1	C Backend . . . . .	108
5.4.2	CPN Hints . . . . .	111
5.5	Case Study . . . . .	113
5.5.1	Synthetic Example . . . . .	113
5.5.2	Audio Filter Application . . . . .	116
5.6	Limitations and Outlook. . . . .	121
5.7	Synopsis. . . . .	122
<b>6</b>	<b>Parallel Code Flow . . . . .</b>	<b>123</b>
6.1	Tool Flow Overview . . . . .	123
6.1.1	The CPN C Compiler . . . . .	124
6.1.2	Tool Flow Components . . . . .	125
6.1.3	Input and Output Modeling . . . . .	126
6.2	Token Logging and KPN Tracing . . . . .	130
6.2.1	KPN Traces . . . . .	130
6.2.2	Sequential Performance Estimation Revisited. . . . .	131
6.3	Building and Evaluating the Model . . . . .	132
6.3.1	Model Construction. . . . .	132
6.3.2	Model Evaluation . . . . .	137
6.4	Best-Effort Mapping and Scheduling . . . . .	139
6.4.1	Trace Graph . . . . .	139
6.4.2	Graph-Based Buffer Sizing. . . . .	141
6.4.3	Heuristics for Setting Scheduler Parameters . . . . .	142
6.4.4	Heuristics for Decoupled Process and Channel Mapping . . . . .	144
6.4.5	Joint Process and Channel Mapping: The GBM Algorithm. . . . .	147
6.4.6	Post-Processing Phase . . . . .	151
6.5	Mapping and Scheduling with Timing Constraints. . . . .	151
6.6	OSIP Backend. . . . .	154
6.6.1	OSIP Generated Code . . . . .	154
6.6.2	Debugging Environment . . . . .	155
6.7	Case Study . . . . .	156
6.7.1	Experimental Setup . . . . .	156
6.7.2	Best-Effort Results . . . . .	159
6.7.3	Results for Real-Time Applications. . . . .	160
6.7.4	Validation on TI's Keystone Platform . . . . .	161
6.8	Limitations and Outlook. . . . .	164
6.9	Synopsis. . . . .	164

<b>7</b>	<b>Extensions for Software Defined Radio</b>	165
7.1	Tool Flow Overview	165
7.1.1	Tool Flow Components	166
7.1.2	Input and Output Modeling	167
7.2	Tracing for Software Defined Radio	169
7.2.1	Flavor Trace Generation	169
7.2.2	Sequential Performance Estimation Revisited	170
7.3	Configuration Matching	172
7.3.1	From Nucleus to Flavors	172
7.3.2	Interface Matching	173
7.4	Mapping and Code Generation	175
7.4.1	Testing Timing Constraints	175
7.4.2	Code Generator Extensions	176
7.5	Case Study	178
7.5.1	Target Application: MIMO OFDM Transceiver	178
7.5.2	Target Platforms and Flavor Libraries	179
7.5.3	Results of the SDR Flow	181
7.5.4	Execution on the Virtual Platform	183
7.6	Limitations and Outlook	184
7.7	Synopsis	185
<b>8</b>	<b>Multi-Application Flow</b>	187
8.1	Tool Flow Overview	187
8.1.1	Tool Flow Components	188
8.1.2	Input and Output Modeling	189
8.2	Scenario Analysis	191
8.2.1	Generating Configurations for Single Applications	192
8.2.2	Pre-Processing	193
8.2.3	Composability Analysis	195
8.2.4	Results Export	198
8.3	Case Study	198
8.3.1	Experimental Setup	199
8.3.2	Scenario Analysis Results	199
8.4	Limitations and Outlook	201
8.5	Synopsis	203
<b>9</b>	<b>Conclusions and Outlook</b>	205
	<b>References</b>	209
	<b>Glossary</b>	225
	<b>Index</b>	229

Programming Heterogeneous MPSoCs

Tool Flows to Close the Software Productivity Gap

Castrillón Mazo, J.; Leupers, R.

2014, XV, 232 p. 73 illus., 65 illus. in color., Hardcover

ISBN: 978-3-319-00674-1