

Contents

PART I: Real Time Systems — Background	1
1 Real Time System Characteristics	3
1.1 Real-time and Reactive Programs	4
2 Formal Program Development Methodologies	9
2.1 Requirement Specification	10
2.1.1 An Example	12
2.2 System Specifications	13
3 Characteristics of Real-Time Languages	17
3.1 Modelling Features of Real-Time Languages	19
3.2 A Look at Classes of Real-Time Languages	22
4 Programming Characteristics of Reactive Systems	25
4.1 Execution of Reactive Programs	26
4.2 Perfect Synchrony Hypothesis	26
4.3 Multiform Notion of Time	27
4.4 Logical Concurrency and Broadcast Communication	27
4.5 Determinism and Causality	28
PART II: Synchronous Languages	29
5 ESTEREL Language: Structure	31
5.1 Top Level Structure	31
5.1.1 Signals and Events	32
5.1.2 Module Instantiation	33
5.2 ESTEREL Statements	34
5.2.1 Data Handling Statements	36
5.2.2 Reactive Statements	36

5.2.3	Derived Statements	41
5.3	Illustrations of ESTEREL Program Behaviour	43
5.4	Causality Problems	45
5.5	A Historical Perspective	46
6	Program Development in ESTEREL	49
6.1	A Simulation Environment	49
6.2	Verification Environment	52
7	Programming Controllers in ESTEREL	55
7.1	Auto Controllers	55
7.1.1	A Very Simple Auto Controller	55
7.1.2	A Complex Controller	56
7.1.3	A Cruise Controller	58
7.1.4	A Train Controller	61
7.1.5	A Mine Pump Controller	63
8	Asynchronous Interaction in ESTEREL	67
9	Futurebus Arbitration Protocol: A Case Study	71
9.1	Arbitration Process	71
9.2	Abstraction of the Protocol	72
9.3	Solution in ESTEREL	74
10	Semantics of ESTEREL	79
10.1	Semantic Structure	79
10.2	Transition Rules	81
10.2.1	Rules for Signal Statement	84
10.3	Illustrative Examples	87
10.4	Discussions	89
10.5	Semantics of Esterel with exec	90
PART III: Other Synchronous Languages		95
11	Synchronous Language LUSTRE	97
11.1	An Overview of LUSTRE	97
11.2	Flows and Streams	97
11.3	Equations, Variables and Expressions	98
11.4	Program Structure	99
11.4.1	Illustrative Example	101

11.5	Arrays in LUSTRE	102
11.6	Further Examples	103
11.6.1	A Very Simple Auto Controller	103
11.6.2	A Complex Controller	103
11.6.3	A Cruise Controller	104
11.6.4	A Train Controller	106
11.6.5	A Mine Pump Controller	107
12	Modelling Time-Triggered Protocol (TTP) in LUSTRE	111
12.1	Time-Triggered Protocol	111
12.1.1	Clock Synchronization	113
12.1.2	Bus Guardian	114
12.2	Modelling TTP in LUSTRE	115
13	Synchronous Language ARGOS	123
13.1	ARGOS Constructs	123
13.2	Illustrative Example	125
13.3	Discussions	128
PART IV: Verification of Synchronous Programs		133
14	Verification of ESTEREL Programs	133
14.1	Transition System Based Verification of ESTEREL Programs	133
14.1.1	Detailed Discussion	134
14.2	ESTEREL Transition System	135
14.2.1	Abstraction and Hiding	136
14.2.2	Observation Equivalence Reduction	137
14.2.3	Context Filtering	139
14.3	Temporal Logic Based Verification	140
14.4	Observer-based Verification	141
14.5	First Order Logic Based Verification	143
15	Observer Based Verification of Simple LUSTRE Programs	145
15.1	A Simple Auto Controller	145
15.2	A Complex Controller	146
15.3	A Cruise Controller	146
15.4	A Train Controller	147
15.5	A Mine Pump Controller	148

PART V: Integration of Synchrony and Asynchrony	151
16 Communicating Reactive Processes	151
16.1 An Overview of CRP	151
16.2 Communicating Reactive Processes: Structure	153
16.2.1 Syntax of CRP	154
16.2.2 Realizing Watchdog Timers in CRP	155
16.3 Behavioural Semantics of CRP	156
16.4 An Illustrative Example: Banker Teller Machine	157
16.5 Implementation of CRP	160
17 Semantics of Communicating Reactive Processes	165
17.1 A Brief Overview of CSP	165
17.2 Translation of CSP to CRP	166
17.3 Cooperation of CRP Nodes	168
17.4 Ready-Trace Semantics of CRP	168
17.5 Ready-Trace Semantics of CSP	168
17.5.1 Semantic Definition	170
17.5.2 Semantics of Parallel Composition	170
17.5.3 Semantics of ‘send’ Action	170
17.5.4 Semantics of ‘receive’ Action	171
17.5.5 Semantics of Assignment Statement	171
17.5.6 Semantics of Sequential Composition	171
17.5.7 Semantics of Guarded Selection	171
17.6 Extracting CSP Ready-trace Semantics from CRP Semantics	172
17.6.1 Behavioural Traces of CRP Programs	172
17.7 Correctness of the Translation	174
17.8 Translation into MEIJE Process Calculus	176
18 Communicating Reactive State Machines	181
18.1 CRSM Constructs	182
18.2 Semantics of CRSM	183
19 Multiclock ESTEREL	187
19.1 Need for a Multiclock Synchronous Paradigm	187
19.2 Informal Introduction	189
19.2.1 Latched Signals	191
19.2.2 Expressions	192
19.2.3 Multiclock ESTEREL Statements	193

19.2.4	Informal Development of Programs in Multiclock ESTEREL	194
19.3	Formal Semantics	197
19.3.1	Specification of Clocks	197
19.4	Embedding CRP	212
19.5	Modelling a VHDL Subset	218
19.6	Discussion	219
20	Modelling Real-Time Systems in ESTEREL	221
20.1	Interpretation of a Global Clock in terms of <code>exec</code>	222
20.2	Modelling Real-Time Requirements in ESTEREL	222
20.2.1	Deadline Specification	222
20.2.2	Periodic Activities	223
20.2.3	Guaranteed Activities	224
21	Putting it Together	231
	Bibliography	235
	Index	243