

09PH
D553



Complexity Management and Modelling of VLSI Systems

Alex Dickinson B.E.(Hons)

A thesis submitted to the Faculty of Engineering
for the Degree of Doctor of Philosophy.

Department of Electrical and Electronic Engineering
The University of Adelaide
South Australia

January 1988

Contents

Abstract	xii
Preface	xiii
Acknowledgements	xiv
1 Introduction	1
1.1 VLSI System Design Complexity	3
1.1.1 Highly Partitioned Systems	5
1.1.2 Functionally Partitioned Systems	6
1.2 Complexity and Structure	7
1.2.1 Hierarchy	9
1.2.2 Regularity	12
1.3 Structured Design	13
1.3.1 Abstraction	14
1.3.2 Modularity	15
1.3.3 Hierarchy	16
1.3.4 Information Hiding	17
1.3.5 Limited Constructs	17
1.3.6 Regularity	18
1.3.7 Design Procedures	20
1.4 Structural Design	21
1.5 Summary	23

2	Structural VLSI Design	25
2.1	Software Partitioning Criteria	25
2.1.1	Coupling	26
2.1.2	Cohesion	28
2.1.2.1	Summary	31
2.2	Structural Software Design	31
2.2.1	Functional Decomposition	31
2.2.2	Data Flow Design	33
2.2.3	Data Structure Design	34
2.2.4	Summary	35
2.3	VLSI Partitioning Issues	36
2.3.1	Large Systems	36
2.3.2	Unstructured Mediums	37
2.3.3	Concurrency	37
2.3.4	Communication	38
2.3.5	Packaging	39
2.3.6	Regularity	40
2.3.6.1	Regularity of Function	40
2.3.6.2	Regularity of Interconnect	41
2.3.7	Performance	41
2.3.8	Discussion	41
2.4	Abstraction and Hierarchical Equivalence	42
2.5	Structural VLSI Design	44
2.5.1	Planning and Construction	45
2.5.2	Design Description	47
2.5.3	Modelling Algorithmic Function	47
2.5.4	Modelling Physical Form	48
2.5.5	Structural Design: Requirements	48
2.6	Related Research	49

2.6.1	Graph Partitioning	49
2.6.2	Stepwise Layout Refinement	49
2.6.3	Partitioning Evaluation	50
2.6.4	Integrated Descriptions	50
2.7	Summary	51
3	A VLSI Description Language	52
3.1	Introduction	52
3.2	Hardware Description Languages	53
3.2.1	Register Transfer Descriptions	53
3.2.2	Token Passing Descriptions	54
3.2.3	Provably Correct Descriptions	55
3.2.4	Generalized Descriptions	56
3.3	Language Requirements	57
3.3.1	Function	57
3.3.2	Structure	57
3.3.3	Communication	58
3.3.4	Reusability	58
3.3.5	Regularity	59
3.3.6	Discussion	59
3.4	A Prototype of the Language	60
3.5	Selection of a Base Language	61
3.6	Structural Description	61
3.6.1	Hierarchy	62
3.6.2	Interconnect	63
3.6.3	Structural Regularity	66
3.7	Specification of Function	67
3.7.1	Declarations	67
3.7.2	Initialization	67
3.7.3	Functional Code	67

3.7.4	Specification of Time and Communication	68
3.7.5	Description of Time	69
3.7.6	Outgoing Communication	69
3.7.7	Incoming Communication	70
3.7.8	Synchronization	70
3.7.9	Bus Operations	71
3.8	Systems, Subsystems and Libraries	71
3.9	Parameterization	73
3.10	Limitations	74
3.11	An Example	75
3.12	Comparison	78
3.12.1	Structural Description	79
3.12.2	Functional Description	80
3.12.3	Timing	80
3.13	Summary	81
4	Modelling Function: Simulation	82
4.1	Introduction	82
4.2	Functional Modelling Requirements	83
4.2.1	Architectural Evaluation	83
4.2.1.1	Quantitative Evaluation	84
4.2.1.2	Qualitative Evaluation	86
4.2.2	Functional Verification	86
4.2.2.1	Informal	88
4.2.2.2	Formal	88
4.2.3	System Integration	88
4.3	Design and Implementation Overview	89
4.3.1	The Implementation Language	90
4.4	Translation to Modula-2	92
4.5	Internal Representations	94

4.5.1	Conventional Representations	94
4.5.2	Definitions	95
4.5.3	Instances	96
4.6	Initialization	97
4.6.1	Genesis	98
4.6.2	Instance Creation	98
4.6.3	Instance Hierarchy Creation	99
4.6.4	Connection Creation	99
4.6.5	Port Declaration	100
4.6.6	Function Creation	101
4.7	Active Simulation	101
4.8	Communication and Scheduling	102
4.8.1	Communication Semantics	102
4.8.2	The Scheduling Problem	103
4.8.3	Interprocess Communication Techniques	104
4.8.4	Demand Driven Scheduling	105
4.8.5	Delay Driven Scheduling	108
4.8.6	Discussion	109
4.9	Software Organization	110
4.10	The Simulator Interface	111
4.10.1	Translation and Compilation	111
4.10.2	The Command Interpreter	112
4.10.3	Browsing the Description Structure	112
4.10.4	Error Detection and Notification	113
4.10.5	System Stimulation	113
4.10.5.1	Interactive	114
4.10.5.2	File Based	114
4.10.5.3	Language Based	115
4.10.6	System Observation	115

4.10.7	Interactive	115
4.10.7.1	File Based	116
4.10.7.2	Language Based	116
4.11	Limitations	117
4.12	Summary	118
5	Partitioning for Physical Form	120
5.1	Introduction	120
5.2	Constraints in Physical Design	121
5.2.1	Connection Length	121
5.2.2	Connection Area	122
5.2.3	Connection Planarity	122
5.3	Physical Partitioning Techniques	123
5.3.1	Highly Partitioned Designs	124
5.3.2	Functionally Partitioned Designs	125
5.4	Structured Floorplans	126
5.5	Creating Structured Floorplans	129
5.5.1	Design Procedures	131
5.5.1.1	Planning	131
5.5.1.2	Sequencing	132
5.5.2	Designing with Rectangles	133
5.5.3	Hierarchical Interactions	134
5.5.4	Structuring Techniques	136
5.5.5	Maintaining Planarity	137
5.5.6	Technology Independence	139
5.6	Evaluating Form with Structured Floorplans	140
5.7	Construction with Structured Floorplans	141
5.8	Summary	142
6	Modelling Form: Floorplanning	144

6.1	Introduction	144
6.2	Approaches to Automatic Floorplanning	145
	6.2.1 Planar Graph Techniques	147
	6.2.2 Slicing Techniques	150
	6.2.3 Knowledge Based Techniques	152
	6.2.1 Discussion	154
6.3	Knowledge Based Design	156
	6.3.1 Knowledge Representation	157
	6.3.1.1 Production Rules	157
	6.3.1.2 Frames	159
	6.3.1.3 Algorithms and Data Structures	159
	6.3.2 Reasoning Techniques	160
	6.3.2.1 Abstraction	161
	6.3.2.2 Planning	161
	6.3.2.3 Constraints and Least Commitment	162
	6.3.2.4 Backtracking	162
	6.3.2.5 Metaknowledge	163
	6.3.2.6 Inexact Reasoning	164
6.4	System Overview	166
6.5	The Spatial Reasoning Subsystem	168
	6.5.1 The Rectangular Graph	169
	6.5.2 Turning	171
	6.5.3 Properties of the Embedded RG	172
	6.5.3.1 Planar Embeddability	172
	6.5.3.2 Side Arcs	173
	6.5.3.3 Triangulation	173
	6.5.4 Ensuring Feasible Placements	177
	6.5.5 Placement Enumeration	181
	6.5.6 The RG Data Structure	182

6.5.6.1	Placement Generation	183
6.5.6.2	Placement Acceptance	184
6.5.7	Triangulation	185
6.5.8	Size Estimation	188
6.5.9	Planarization Support	189
6.5.10	Interface Definition	190
6.6	The Classes Subsystem	190
6.7	The Paths Subsystem	193
6.8	The Simplex Subsystem	195
6.9	The Sketcher Subsystem	196
6.10	The Design Manager	198
6.10.1	Control Strategies	199
6.10.2	Initialization	201
6.10.3	Planning	203
6.10.4	Wait for Placements	203
6.10.5	Placement Filtering	204
6.10.6	Module Selection	205
6.10.7	Placement Selection	205
6.10.8	Placement Acceptance	208
6.10.9	Constraint Propagation	208
6.10.10	Deferred Tasks	209
6.10.11	Planarization Fault Handling	209
6.10.12	Route Completion	210
6.10.13	Solve	210
6.11	Implementation	211
6.11.1	Design Manager Implementation	213
6.11.1.1	The Working Memory	213
6.11.1.2	The Rule Set	214
6.11.1.3	The Rule Interpreter	216

6.11.2	Subsystem Implementations	217
6.11.2.1	Symbols	217
6.11.2.2	Lists	219
6.11.2.3	Automatic Memory Management	219
6.11.2.4	Macros	219
6.11.3	Subsystem Interface Mechanisms	221
6.12	Limitations	221
6.13	Summary	222
7	A Structural Design Case Study	224
7.1	Introduction	224
7.2	A Design Example	225
7.3	Functional Overview	226
7.4	An Initial Partitioning	226
7.4.1	Specification	228
7.4.1.1	The Alignment Unit	229
7.4.1.2	The Mantissa Adder	230
7.4.1.3	The Adjuster	230
7.4.1.4	The FPA Composition	232
7.4.2	Modelling Function	233
7.4.3	Modelling Form	234
7.5	Discussion and Modified Partitioning	235
7.6	Refining the Exponent Adder	238
7.7	Refining the Adder	241
7.8	Summary	245
8	Conclusion	246
	References	249

Abstract

The major objective of the research described in this thesis is to describe effective methods for the partitioning of Very Large Scale Integrated (VLSI) systems.

A comparison is made between the structural design of large programs and large VLSI designs. A methodology for VLSI structural design is proposed based on many of the precepts of structural program design. The methodology requires restructuring of the design process; a specific form of design representation; and the addition of computer aided modelling of both the algorithmic function and geometrical form of the structural design.

The design process is divided into a designer intensive top-down planning phase, and an automatic bottom-up construction phase.

A model of design is described in which the commonly used layered set of abstractions are replaced by a single structural description. The language introduced for this purpose incorporates a number of features to reduce the apparent complexity of the system description. The abstract representation of intermodule synchronization and communication allows a process of stepwise refinement to be applied to these as well as the more usual structural entities.

Two computer aided modelling tools have been developed that together constitute a facility for rapidly analyzing alternate structural partitionings in the search for an acceptable design.

The first computer aided modelling tool is a functional simulator incorporating a novel interprocess communication and scheduling mechanism. This allows the efficient implementation of the description language intermodule communication semantics. The interactive nature of the simulator facilitates initial debugging and qualitative evaluation of the design. A profiling facility allows for the quantitative evaluation of the partitioning based on data flow and module activity.

The second computer aided modelling tool is a hierarchical floorplanner that facilitates the evaluation of the embedding of the proposed structural design into the plane. An investigation of structured floorplan design shows that the process is inherently knowledge intensive, and that much of that knowledge is inexact. The floorplanner incorporates several novel knowledge representations that are used to express diverse classes of designer expertise. A representation for spatial reasoning provides for the

efficient manipulation of rectangles in a mosaic. Another representation has been developed for reasoning with the inexact knowledge used by designers in predicting the implementation of floorplan modules in a hierarchical design. A production system is used as a design manager to guide the overall development of the design.

A case study is presented that demonstrates the utility of the methodology and computer aided design tools in VLSI system design.