

# Contents

|          |                                                                                                                       |          |
|----------|-----------------------------------------------------------------------------------------------------------------------|----------|
| <b>1</b> | <b>Introduction</b>                                                                                                   | <b>1</b> |
| 1.1      | Related Work . . . . .                                                                                                | 3        |
| 1.1.1    | Comparison with ISA . . . . .                                                                                         | 3        |
| 1.1.1.1  | Use of Logic and Abstraction Functions . . . . .                                                                      | 3        |
| 1.1.1.2  | Theorem Proving . . . . .                                                                                             | 6        |
| 1.1.1.3  | Term Rewriting Systems . . . . .                                                                                      | 8        |
| 1.1.1.4  | ADL Driven Pipeline Validation . . . . .                                                                              | 11       |
| 1.1.1.5  | FSM Based Methods . . . . .                                                                                           | 12       |
| 1.1.1.6  | Other Methodologies . . . . .                                                                                         | 12       |
| 1.1.2    | Assertion Based Verification . . . . .                                                                                | 14       |
| 1.1.2.1  | Benefits of ABV . . . . .                                                                                             | 14       |
| 1.1.3    | Languages for Assertion Based Verification . . . . .                                                                  | 15       |
| 1.1.3.1  | LTL - Linear Temporal Logic . . . . .                                                                                 | 15       |
| 1.1.3.2  | Applications and Extensions . . . . .                                                                                 | 17       |
| 1.1.3.3  | System Verilog Assertions . . . . .                                                                                   | 17       |
| 1.1.3.4  | Data-Orientation in SVA . . . . .                                                                                     | 19       |
| 1.1.3.5  | Dynamic Property Verification . . . . .                                                                               | 21       |
| 1.1.3.6  | Formal Property Verification . . . . .                                                                                | 22       |
| 1.1.4    | Test Generation . . . . .                                                                                             | 23       |
| 1.1.4.1  | Graph Model Based Test Generation . . . . .                                                                           | 24       |
| 1.1.4.2  | Specification Language Driven Test Generation . . . . .                                                               | 25       |
| 1.1.4.3  | SAT based methods . . . . .                                                                                           | 26       |
| 1.1.4.4  | Property Checking based methods . . . . .                                                                             | 27       |
| 1.1.4.5  | FSM based methods . . . . .                                                                                           | 27       |
| 1.1.4.6  | Other methods . . . . .                                                                                               | 28       |
| 1.2      | Motivation and Objectives . . . . .                                                                                   | 29       |
| 1.2.1    | Trace Equivalence: . . . . .                                                                                          | 30       |
| 1.2.2    | Assertion Based Verification of Pipeline: . . . . .                                                                   | 31       |
| 1.2.3    | Test Generation . . . . .                                                                                             | 32       |
| 1.3      | Contribution of this Thesis . . . . .                                                                                 | 33       |
| 1.3.1    | A systematic framework for Validation and Debugging of Pipelined<br>Simulators . . . . .                              | 33       |
| 1.3.2    | Simulation Based Verification using Temporally Attributed Boolean<br>Logic . . . . .                                  | 35       |
| 1.3.3    | A Framework for Scenario Driven Test Case Generation for<br>Functional Verification of Pipelined Processors . . . . . | 37       |
| 1.4      | Thesis Organization . . . . .                                                                                         | 38       |

|          |                                                                                   |           |
|----------|-----------------------------------------------------------------------------------|-----------|
| <b>2</b> | <b>A Framework for Systematic Validation and Debugging of Pipeline Simulators</b> | <b>39</b> |
| 2.1      | Problem Formulation and Definitions                                               | 41        |
| 2.1.1    | Notations                                                                         | 41        |
| 2.1.2    | Problem Formulation                                                               | 42        |
| 2.1.2.1  | The Model                                                                         | 42        |
| 2.1.2.2  | Assumptions                                                                       | 44        |
| 2.1.2.3  | Modeling Special Pipeline Techniques                                              | 44        |
| 2.1.2.4  | $D^*$ equivalence                                                                 | 45        |
| 2.2      | Proposed Algorithm and Analysis                                                   | 46        |
| 2.2.1    | The Algorithm                                                                     | 46        |
| 2.2.2    | Proof of Correctness                                                              | 52        |
| 2.2.3    | Complexity Analysis                                                               | 54        |
| 2.3      | Examples                                                                          | 55        |
| 2.4      | Adaptations                                                                       | 58        |
| 2.4.1    | Runtime Mode                                                                      | 59        |
| 2.4.2    | Register Verification                                                             | 62        |
| 2.4.3    | Temporary Buffering of Register Values                                            | 62        |
| 2.4.4    | Pipelines Where Incorrectly Predicted Instructions Access Registers               | 63        |
| 2.4.5    | Handling Certain Internal Semantics of Instructions                               | 63        |
| 2.5      | Implementation and Results                                                        | 64        |
| 2.5.1    | Experimental Setup                                                                | 64        |
| 2.5.2    | Validation of Pipeline Simulator                                                  | 65        |
| 2.5.3    | Fault Detection                                                                   | 66        |
| 2.5.3.1  | Bug A                                                                             | 66        |
| 2.5.3.2  | Bug B                                                                             | 68        |
| 2.5.3.3  | Bug C                                                                             | 68        |
| 2.5.3.4  | Bug D                                                                             | 69        |
| 2.5.4    | Debugging Through Faults                                                          | 69        |
| 2.6      | Summary                                                                           | 71        |
| <b>3</b> | <b>Simulation Based Verification Using Temporally Attributed Boolean Logic</b>    | <b>73</b> |
| 3.1      | Introduction                                                                      | 73        |
| 3.2      | TAB Logic Specifications                                                          | 77        |
| 3.2.1    | Syntax of TAB Logic                                                               | 77        |
| 3.2.2    | Temporal and TAB Expression                                                       | 78        |
| 3.2.2.1  | Temporal Expressions:                                                             | 80        |
| 3.2.2.2  | Temporally Attributed Boolean (TAB) Expressions:                                  | 80        |
| 3.2.3    | Semantics of Temporal Expressions                                                 | 81        |
| 3.2.4    | Examples                                                                          | 83        |
| 3.2.5    | A nested TAB Logic Specification                                                  | 86        |
| 3.3      | Algorithms for Checking TAB Specifications                                        | 87        |
| 3.3.1    | Temporal System of Equations (TSE)                                                | 87        |
| 3.3.2    | Temporal System Graph                                                             | 88        |
| 3.3.3    | An Offline Algorithm for Checking TSE with An Acyclic TSG                         | 88        |
| 3.3.3.1  | Correctness of the Offline Algorithm                                              | 90        |

|         |                                                                                                                               |            |
|---------|-------------------------------------------------------------------------------------------------------------------------------|------------|
| 3.3.3.2 | Example Trace of the Offline Algorithm . . . . .                                                                              | 92         |
| 3.3.4   | An Online Algorithm for Checking TSE with An Acyclic TSG . . . . .                                                            | 93         |
| 3.3.5   | Steps of the Online Algorithm . . . . .                                                                                       | 98         |
| 3.3.5.1 | Evaluation of Temporal and TAB Expressions . . . . .                                                                          | 100        |
| 3.3.5.2 | Updating the Frontiers . . . . .                                                                                              | 101        |
| 3.3.5.3 | Simulation History and Records Deletion . . . . .                                                                             | 102        |
| 3.4     | Extensions to TAB Logic . . . . .                                                                                             | 104        |
| 3.4.1   | Evaluation Instant Based Extension . . . . .                                                                                  | 104        |
| 3.4.2   | Counter Based Extension . . . . .                                                                                             | 108        |
| 3.5     | Expressiveness of TAB Logic . . . . .                                                                                         | 109        |
| 3.5.1   | TAB Logic with Next Operator on an Infinite Trace . . . . .                                                                   | 109        |
| 3.5.1.1 | Redefining the Next operator . . . . .                                                                                        | 109        |
| 3.5.1.2 | Representation of LTL expressions by TAB logic as-<br>sertions . . . . .                                                      | 110        |
| 3.5.1.3 | Motivation for TAB logic with nested assertions . . . . .                                                                     | 110        |
| 3.6     | Case Study - Processor Implementation Verification . . . . .                                                                  | 112        |
| 3.6.1   | Architecture of the System . . . . .                                                                                          | 112        |
| 3.6.2   | Instruction Semantics Verification . . . . .                                                                                  | 113        |
| 3.6.2.1 | TAB specification for <i>loadw</i> instruction . . . . .                                                                      | 114        |
| 3.6.2.2 | TAB specification for <i>storw</i> instruction . . . . .                                                                      | 115        |
| 3.6.2.3 | TAB specification for <i>tbitw</i> instruction . . . . .                                                                      | 115        |
| 3.6.2.4 | TAB specification for <i>jump</i> instruction . . . . .                                                                       | 116        |
| 3.6.2.5 | TAB specification for <i>jovf</i> instruction . . . . .                                                                       | 116        |
| 3.6.2.6 | TAB specification for <i>cinw</i> instruction . . . . .                                                                       | 116        |
| 3.6.2.7 | SVA description of the loadw instruction - A comparison . . . . .                                                             | 117        |
| 3.6.3   | Bus Transaction Verification . . . . .                                                                                        | 118        |
| 3.6.3.1 | Priority of Bus Transactions: . . . . .                                                                                       | 118        |
| 3.6.3.2 | TAB specification for Priority of Bus Transactions . . . . .                                                                  | 119        |
| 3.6.4   | Interrupt and Exception Mode Verification . . . . .                                                                           | 119        |
| 3.6.4.1 | TAB specification for Interrupt Processing . . . . .                                                                          | 120        |
| 3.7     | Experimentation and Results . . . . .                                                                                         | 122        |
| 3.7.1   | Experimental Setup . . . . .                                                                                                  | 122        |
| 3.7.2   | Validation of the Implementation . . . . .                                                                                    | 123        |
| 3.7.2.1 | Interrupt Behavior Validation . . . . .                                                                                       | 124        |
| 3.7.3   | Fault Detection . . . . .                                                                                                     | 125        |
| 3.7.3.1 | Fault A: . . . . .                                                                                                            | 125        |
| 3.7.3.2 | Fault B: . . . . .                                                                                                            | 126        |
| 3.7.3.3 | Fault C: . . . . .                                                                                                            | 126        |
| 3.8     | Case Study - Simulation Based Validation of Analog Circuits . . . . .                                                         | 127        |
| 3.8.1   | Frequency Analysis of Operational Amplifier . . . . .                                                                         | 128        |
| 3.8.2   | Stability of DC-DC Converter . . . . .                                                                                        | 128        |
| 3.9     | Limitations of TAB Logic . . . . .                                                                                            | 132        |
| 3.10    | Summary . . . . .                                                                                                             | 133        |
| 4       | <b>A Framework for Scenario Driven Test Case Generation for Func-<br/>tional Validation of Pipelined Processors</b> . . . . . | <b>135</b> |
| 4.1     | Problem Formulation and Definitions . . . . .                                                                                 | 139        |
| 4.1.1   | The Test Generation Framework . . . . .                                                                                       | 140        |

|         |                                                                            |     |
|---------|----------------------------------------------------------------------------|-----|
| 4.1.2   | Testcases . . . . .                                                        | 141 |
| 4.1.2.1 | Hazard Free and Hazard Based Test Case . . . . .                           | 141 |
| 4.1.3   | Pipeline Model . . . . .                                                   | 141 |
| 4.1.3.1 | Instruction Set Architecture (ISA) . . . . .                               | 143 |
| 4.1.3.2 | Pipeline Behavior Specification . . . . .                                  | 143 |
| 4.1.4   | Scenario Description Language . . . . .                                    | 144 |
| 4.1.4.1 | Syntax of the Scenario Description Language . . . . .                      | 144 |
| 4.1.4.2 | Semantics of Scenario Description Language . . . . .                       | 145 |
| 4.2     | Example Description of Scenarios . . . . .                                 | 146 |
| 4.2.1   | Description of Standard Scenarios . . . . .                                | 147 |
| 4.2.2   | Insufficient Ports Hazard . . . . .                                        | 148 |
| 4.2.3   | Exceptions . . . . .                                                       | 148 |
| 4.2.4   | Predictive Stalling . . . . .                                              | 148 |
| 4.3     | Pipeline State Generation . . . . .                                        | 149 |
| 4.3.1   | Pipeline State . . . . .                                                   | 150 |
| 4.3.2   | Predicate Table Generation . . . . .                                       | 150 |
| 4.3.2.1 | The <i>before</i> ( $X, I, J$ ) Predicate . . . . .                        | 151 |
| 4.3.3   | Variable Instantiation . . . . .                                           | 151 |
| 4.3.4   | An Example . . . . .                                                       | 152 |
| 4.4     | Hazard Free Test Case Generation . . . . .                                 | 153 |
| 4.4.1   | Test Generation Algorithm . . . . .                                        | 154 |
| 4.4.2   | Example Run - RAW Hazard Test case . . . . .                               | 155 |
| 4.4.3   | Complexity Analysis of the Hazard Free Test Generation Algorithm . . . . . | 155 |
| 4.5     | SAT based Test Generation . . . . .                                        | 156 |
| 4.5.1   | Formulation of the Pipeline behavior using propositions . . . . .          | 157 |
| 4.5.2   | Pipeline Constraints Modeled as Propositions . . . . .                     | 159 |
| 4.5.2.1 | Execution Flow Constraints . . . . .                                       | 159 |
| 4.5.2.2 | Data Flow Constraints . . . . .                                            | 160 |
| 4.5.2.3 | Resource Constraints . . . . .                                             | 161 |
| 4.5.3   | Hazards . . . . .                                                          | 161 |
| 4.5.3.1 | Structural Hazards . . . . .                                               | 161 |
| 4.5.3.2 | Multicycle Hazards . . . . .                                               | 162 |
| 4.5.3.3 | Implicit Hazards . . . . .                                                 | 162 |
| 4.5.3.4 | Data Hazards . . . . .                                                     | 162 |
| 4.5.4   | VLIW Constraints . . . . .                                                 | 163 |
| 4.5.5   | Propositions of the Pipeline State . . . . .                               | 163 |
| 4.5.5.1 | Proposition Generation . . . . .                                           | 164 |
| 4.5.5.2 | All Test Cases . . . . .                                                   | 165 |
| 4.5.5.3 | Minimal Test case . . . . .                                                | 165 |
| 4.5.6   | Pipeline Clauses and Test Generation . . . . .                             | 165 |
| 4.5.7   | Optimization . . . . .                                                     | 166 |
| 4.5.7.1 | Optimization of 'E'-Clauses . . . . .                                      | 166 |
| 4.5.7.2 | Optimization of 'H'-Clauses . . . . .                                      | 167 |
| 4.5.7.3 | Optimization of State-Clauses . . . . .                                    | 168 |
| 4.5.8   | Modeling Special Pipeline Behavior . . . . .                               | 168 |
| 4.5.8.1 | Forwarding . . . . .                                                       | 169 |
| 4.5.8.2 | Multiple Execution Units . . . . .                                         | 169 |
| 4.5.9   | Example Test Program Generated by the SAT approach . . . . .               | 170 |

|          |                                                                 |            |
|----------|-----------------------------------------------------------------|------------|
| 4.6      | Implementation and Results . . . . .                            | 171        |
| 4.6.1    | Experimental Setup . . . . .                                    | 171        |
| 4.6.2    | Pipeline Configurations . . . . .                               | 171        |
| 4.6.3    | Hazard Free Test Generation Results . . . . .                   | 173        |
| 4.6.4    | SAT Based Test Generation Results . . . . .                     | 173        |
| 4.6.5    | Comparison of the Approaches . . . . .                          | 175        |
| 4.7      | Summary . . . . .                                               | 176        |
| <b>5</b> | <b>Conclusions and Future Work</b>                              | <b>177</b> |
| 5.1      | Summary of Contributions . . . . .                              | 177        |
| 5.2      | Scope of Future Research . . . . .                              | 178        |
| 5.2.1    | Extensions to $D^*$ Equivalence . . . . .                       | 178        |
| 5.2.2    | Extensions to TAB Logic - Software Verification . . . . .       | 179        |
| 5.2.3    | Model Checking TAB Logic . . . . .                              | 179        |
| 5.2.4    | Exploring Complex Pipeline Features - Test Generation . . . . . | 179        |
|          | <b>Bibliography</b>                                             | <b>183</b> |