TABLE OF CONTENTS

1 INTRODUCTION 1

2 DEFINITIONS AND TAXONOMY OF COMPUTER ARCHITECTURES FOR MACHINE EMBEDDED CONTROL SYSTEMS 7

2.1 Control System 7
2.2 Embedded Control System 8
2.3 Machine Embedded Control System 10
2.4 Computer Architecture for a Machine Embedded Control System 13
2.5 Taxonomy of Different Computer Architectures 14
2.5.1 Flynn Taxonomy 14
2.5.2 Anderson Taxonomy 15
2.5.3 Händler ECS Taxonomy 17
2.5.4 Giloi Taxonomy 19
2.5.5 Extensions and Refinement of Giloi’s Taxonomy 21
2.5.5.1 Classification of the Communication Structures 21
2.5.5.2 Control Structure Extension and Refinement 45
2.5.6 RISC/CISC Taxonomy 47
2.5.7 Other Taxonomies of Specifically Dataflow Architectures 48
2.6 Conclusions of Chapter 2 48
3 COMPUTER ARCHITECTURES DEPLOYED IN MACHINE EMBEDDED CONTROL SYSTEMS 49

3.1 Hardware Structures 50
3.1.1 Communication Structure 50
3.1.2 Processing Element (PE) Structures 52
3.1.2.1 Data and Instruction Path Organization 53
3.1.2.1.1 von Neumann-type Structures 53
3.1.2.1.2 Harvard Structure 55
3.1.2.2 Functional Complexity 55
3.1.2.2.1 Specialized Simple ALU PEs 55
3.1.2.2.2 Multifunctional ALU PE 56
3.2 Operational Principle 58
3.2.1 Set of Operations 58
3.2.1.1 CISC 58
3.2.1.2 RISC 59
3.2.2 Execution Control 60
3.2.2.1 Control Flow 60
3.2.2.1.1 Sequential (von Neumann) Control Flow 60
3.2.2.1.2 Parallel (Multi-thread) Execution 62
3.2.2.1.3 Multiflow Execution 63
3.2.2.2 Dataflow Driven Execution 65
3.2.2.2.1 Static Dataflow Driven Execution 66
3.2.2.2.2 Dynamic Dataflow Driven Execution 67
3.2.2.3 Process level Dataflow Driven Execution 69
3.2.2.4 Demand-Driven (Reduction) Execution 71

4 PERFORMANCE ANALYSIS OF COMPUTER ARCHITECTURES DEPLOYED IN MACHINE EMBEDDED CONTROL SYSTEMS 73

4.1 Evaluation Criterion 74
4.2 Modelling Approximations and Assumptions 75
4.3 Queueing Model of Control Flow Driven Architectures 76
4.4 Queueing Model of Dataflow Driven Architectures 79
4.4.1 Mean Waiting Time Spent in a Matching Section 80
4.4.2 Mean Time Spent in a Waiting Queue 82
4.5 Evaluation of the Results 86
## Table of Contents

### 5 COMPUTER OPERATION IN TERMS OF TOKENS

5.1 Tokens and Operations 91
5.1.1 Types of Tokens 91
5.1.2 Basic Actions Concerning Tokens 95
5.1.3 The Operations 95
5.2 Vanishing and Replication of Tokens 98
5.2.1 Single-Entry-Single-Exit Operations 98
5.2.2 Multiple-Entry-Single-Exit Operations 99
5.2.3 Single-Entry-Multiple-Exit Operations 101
5.2.4 Multiple-Entry-Multiple-Exit Operations 103
5.3 Requirements to be Fulfilled by an Embedded Control System 104
5.3.1 Global Invariance Criterion 104
5.3.1.1 Introduction and Definition 104
5.3.1.2 Design Methods that Satisfy the Global Invariance Criterion 105
5.3.1.2.1 Methods Based on the Satisfaction of the Serialization Principle and Deterministic Sequential Proof 105
5.3.1.2.2 Methods Based on Dijkstra Calculus for the Program Derivation 106
5.3.1.2.3 Methods Based on Hoare's Trace Theory 107
5.3.1.2.4 Systems Verified with Rem's Trace Theory 107
5.3.2 Liveness and Livelock Free Systems 108
5.3.3 Deadlock Free Operation 109
5.3.4 Deterministic Operation 111
5.3.4.1 Basic Problem of Nondeterminism 111
5.3.4.2 Sources of Potential Nondeterminism in Computer Architecture 114

### 6 OPERATIONAL PRINCIPLES FOR OR TOKENS

6.1 OR Tokens in Control and Dataflow 117
6.1.1 OR and XOR Relation Between the Output Tokens 118
6.1.2 OR Input Token Relation 121
6.1.2.1 OR Input Tokens in Control Flow Driven Architectures 121
6.1.2.2 OR Input Token in Dataflow Architectures 123
6.2 OR Dataflow on the Process Level 127
6.2.1 An Overview 127
6.2.2 Internal Process Execution Trigger 128
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.3</td>
<td>Relation Between the Output Tokens</td>
<td>129</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Feasibility of Operation</td>
<td>130</td>
</tr>
<tr>
<td>6.2.4.1</td>
<td>Description of a Selected Example</td>
<td>130</td>
</tr>
<tr>
<td>6.2.4.2</td>
<td>Example of Dataflow Implementation - Version 1</td>
<td>132</td>
</tr>
<tr>
<td>6.2.4.3</td>
<td>Example of Dataflow Implementation - Version 2</td>
<td>132</td>
</tr>
<tr>
<td>6.2.4.4</td>
<td>Example of Dataflow Implementation - Version 3</td>
<td>134</td>
</tr>
<tr>
<td>6.2.4.5</td>
<td>Example of Dataflow Implementation - Version 4</td>
<td>135</td>
</tr>
<tr>
<td>6.3</td>
<td>Methods of Increasing the Efficiency of OR Dataflow Architectures</td>
<td>136</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Output Tokens Reduction Mechanism</td>
<td>136</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Lazy Evaluation and Cancellation of Computations</td>
<td>138</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Bounded Propagation of Tokens Along Data Paths</td>
<td>140</td>
</tr>
<tr>
<td>6.4</td>
<td>OR Dataflow Architecture Operational Principle</td>
<td>141</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Process Characteristics</td>
<td>141</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Identification of the Activities of an OR Process Level Dataflow</td>
<td>143</td>
</tr>
<tr>
<td>6.4.2.1</td>
<td>Input Data Identification</td>
<td>143</td>
</tr>
<tr>
<td>6.4.2.2</td>
<td>Process Assignment</td>
<td>144</td>
</tr>
<tr>
<td>6.4.2.3</td>
<td>Router</td>
<td>145</td>
</tr>
<tr>
<td>6.4.2.4</td>
<td>Linker and Execution</td>
<td>146</td>
</tr>
<tr>
<td>6.4.2.5</td>
<td>Output Setting</td>
<td>146</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Concurrency of Activities</td>
<td>147</td>
</tr>
<tr>
<td>6.4.3.1</td>
<td>Theoretically Feasible Concurrency</td>
<td>147</td>
</tr>
<tr>
<td>6.4.3.2</td>
<td>Effective Concurrency</td>
<td>150</td>
</tr>
<tr>
<td>6.4.3.2.1</td>
<td>INPUT_TOKEN and incoming I-RES_TOKEN</td>
<td>150</td>
</tr>
<tr>
<td>6.4.3.2.2</td>
<td>IN_DATA_TOKEN</td>
<td>150</td>
</tr>
<tr>
<td>6.4.3.2.3</td>
<td>PROCESS_TOKEN</td>
<td>151</td>
</tr>
<tr>
<td>6.4.3.2.4</td>
<td>EXECUTOR_TOKEN</td>
<td>151</td>
</tr>
<tr>
<td>6.4.3.2.5</td>
<td>LOCAL_TOKEN</td>
<td>151</td>
</tr>
<tr>
<td>6.4.3.2.6</td>
<td>OUT_DATA_TOKEN</td>
<td>152</td>
</tr>
<tr>
<td>6.4.3.2.7</td>
<td>OUTPUT_TOKEN And Outgoing I-RES_TOKEN</td>
<td>152</td>
</tr>
<tr>
<td>6.4.4</td>
<td>Execution Examples</td>
<td>154</td>
</tr>
<tr>
<td>6.4.4.1</td>
<td>Token Conversion During Execution Caused by \texttt{.\texttt{reset}} in Dataflow Version 3</td>
<td>154</td>
</tr>
<tr>
<td>6.4.4.2</td>
<td>Token Conversion During Execution Caused by Axis Input Data Y1 in Dataflow Version 3</td>
<td>156</td>
</tr>
<tr>
<td>6.4.4.3</td>
<td>Token Conversion During Execution Caused by Mode Change in Dataflow Version 4</td>
<td>157</td>
</tr>
</tbody>
</table>
7 HARDWARE STRUCTURE SUPPORTING THE OR DATAFLOW OPERATIONAL PRINCIPLE 159

7.1 Communication Structure 160
7.1.1 Communication Structure Alternatives 160
7.1.1.1 Dedicated Communication Structure 160
7.1.1.2 Bus Communication 161
7.1.1.3 Multiple Bus 162
7.1.1.4 Cube and Hypercube 164
7.1.2 Distributed Versus Centralized Assignment and Routing 166
7.1.3 Detailed Evaluation of Bus Based Communication Structures 166
7.1.3.1 Modelling Approximations and Assumptions 167
7.1.3.2 Queueing Model of a Single Bus Based Cluster 171
7.1.3.3 Queueing Model of a Multiple Bus Cluster 175
7.1.3.4 Validation of the Theoretical Results by an Analysis of the Execution Times in Comparable Conventional Machine Control Systems 178
7.1.3.4.1 Example of a Welding Robot Slave 2 Executing Unit 178
7.1.3.4.2 Example of a Conventional Grinding Machine Embedded Control System 181
7.1.4 The Selected Communication Structure 184
7.1.4.1 Main Components of the Structure 184
7.1.4.2 Distributor/Collector Designs 185
7.1.4.3 Distributor-Collector Communication 187
7.1.4.4 Input Modules 189
7.1.4.5 Human Operator Interface 189
7.2 Processing Element Structure 191
7.2.1 Basic Components 191
7.2.2 Communication Components 192
7.2.2.1 Distributor - PE Interface 192
7.2.2.2 PE - Collector Interface 193
7.2.2.3 High Rate Serial Interface (Human Operator Interface) 194
7.3 Closing Remarks 196
# Table of Contents

8 OR DATAFLOW OPERATING SYSTEM  

8.1 Functions of the Dataflow Operating System  
8.2 Process and Data Management in Dataflow Operating Systems  
8.2.1 Process and Dataflow Management  
8.2.2 Memory Management in Dataflow Architectures  
8.2.3 Handling of Structures in Dataflow Architectures  
8.2.3.1 Subdivision into Smaller Entities  
8.2.3.2 Macro-Actors  
8.2.3.3 Token Relabelling  
8.2.3.4 Structure Handling Dedicated Operators  
8.2.3.5 Structure Dedicated PE Server  
8.3 Object Identification  
8.3.1 Notation Convention  
8.3.2 Input Data Identification and Process Assignment Objects  
8.3.3 Router Objects  
8.3.4 Linker Objects  
8.3.5 Executor Objects  
8.3.6 Results Collector Objects  
8.4 Operating System Software  
8.4.1 Executional Assumptions  
8.4.2 First Version of the Operating System Software  
8.4.3 Feasible Performance Improvements in the First Version of the Operating System Software  
8.5 Compliance with the Requirements for the Embedded Control System's Computer Operation  
8.5.1 Deadlock  
8.5.2 Liveness and Livelock Free Operation  
8.5.3 Deterministic Operation  
8.5.4 Reliability  
8.5.4.1 Architectural Provisions for Graceful Degradation  
8.5.4.2 Functional Software Provisions for Graceful Degradation  
8.6 Debugging and Maintenance
9 PROGRAMMABILITY

9.1 Programming Language
9.1.1 Procedural Languages
9.1.2 Declarative Languages
9.1.2.1 Dataflow Languages
9.1.2.2 Functional Languages
9.1.2.3 Logical Languages
9.2 Programming Style
9.2.1 Single Assignment
9.2.2 Functional Programming
9.2.3 Graph Based Dataflow Programming
9.3 Programming Style and Language Selected for OR Dataflow Architecture
9.3.1 Programming Style
9.3.2 Programming Language
9.3.2.1 General Considerations
9.3.2.2 An Example

10 MACHINE CONTROL SYSTEM BASED ON OR DATAFLOW COMPUTER ARCHITECTURE

10.1 Control System Identification
10.2 Environment
10.3 Process
10.3.1 Functional Specification
10.3.2 Actuators and Control Parameters
10.4 OR Dataflow Control Computer
10.4.1 Hardware Configuration
10.4.2 Operating System Software
10.4.3 Control Software
10.4.3.1 Input Processes
10.4.3.2 Algorithm Processes
10.4.3.3 Output Processes
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.5</td>
<td>Operator Interface</td>
<td>274</td>
</tr>
<tr>
<td>10.6</td>
<td>Verification of the Control System Performance</td>
<td>275</td>
</tr>
<tr>
<td>10.6.1</td>
<td>Execution Times of the OR Dataflow Control System</td>
<td>275</td>
</tr>
<tr>
<td>10.6.2</td>
<td>Comparison with the Conventional Control Flow Driven System</td>
<td>277</td>
</tr>
<tr>
<td>10.7</td>
<td>Test Evaluation</td>
<td>279</td>
</tr>
<tr>
<td>10.7.1</td>
<td>Test Criteria</td>
<td>279</td>
</tr>
<tr>
<td>10.7.2</td>
<td>Test Bed</td>
<td>280</td>
</tr>
<tr>
<td>10.7.3</td>
<td>Results</td>
<td>281</td>
</tr>
<tr>
<td>10.7.3.1</td>
<td>Cone Grinding</td>
<td>281</td>
</tr>
<tr>
<td>10.7.3.2</td>
<td>The Ground Surface Quality Factor</td>
<td>281</td>
</tr>
<tr>
<td>10.7.3.3</td>
<td>Roundness of a Cylindrically Ground Surface</td>
<td>282</td>
</tr>
<tr>
<td>10.7.3.4</td>
<td>Surface Straightness Along the XZ Coordinates</td>
<td>284</td>
</tr>
<tr>
<td>11</td>
<td>CONCLUSIONS</td>
<td>287</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
<td>293</td>
</tr>
<tr>
<td></td>
<td>INDEX</td>
<td>309</td>
</tr>
</tbody>
</table>