Contents

List of Figures **xiii**

List of Tables **xix**

Abbreviations and Acronyms **xxi**

Abstract **xxiii**

Resumen **xxv**

Resum **xxvii**

1 Introduction **1**

1.1 Contributions of the Thesis . . . 4

1.2 Thesis Outline . . . 6

2 Background, Related Work and Methodology **9**

2.1 Background . . . 10

2.1.1 NoCs . . . 10

2.1.1.1 Switch Architecture . . . 12

2.1.1.2 Flow Control . . . 13

2.1.1.3 Switching . . . 15

2.1.1.3.1 Virtual Channels (VC) . . . 16

2.1.1.4 Routing Algorithm . . . 17

2.1.1.5 Arbitration . . . 20

2.1.2 Real-Time Systems . . . 20

2.1.2.1 Safety-Critical Real-Time Systems . . . 21

2.1.2.2 Worst Case Execution Time (WCET) Estimates . . . 21

2.1.2.3 Time Division Multiple Access (TDMA) . . . 23

2.1.2.4 Real-Time NoCs . . . 23

2.2 Related Work . . . 26

2.2.1 State-Of-The-Art COTS NoCs and Real-Time NoCs . . . 26

2.3 Methodology . . . 28

2.3.1 Design Process . . . 28

2.3.2 Evaluation Process . . . 29

2.3.2.1 Performance Evaluation . . . 29

2.3.2.2 Implementation Evaluation . . . 30

2.3.2.3 Cycle Accurate Simulation . . . 31

2.3.3 PEAK Architecture . . . 31

3 DCFNoC: A Delayed Conflict-Free Time Division Multiplexing Network on Chip 35

3.1 Delayed Conflict-Free Network . . . 37

3.1.1 Definitions . . . 37

3.2 Building DCFNoC out of layered CDG . . . 43

3.3 Algorithms for the DCFNoC Design Methodology . . . 47

3.3.1 CDG_l Algorithm . . . 47

3.3.2 CDG_{dl} Algorithm . . . 47

3.4 Flexible Bandwidth Allocation . . . 49

3.5 Broadcast Support . . . 50

3.6 Router Design . . . 51

3.7 Performance Evaluation . . . 53

3.7.1 Experimental Setup . . . 53

3.7.2 Timing Guarantees . . . 54

3.7.3 Flexible Bandwidth Allocation . . . 56

3.7.4 Area and Frequency . . . 57

3.8 Summary . . . 58

4 Enforcing Predictability of manycores with DCFNoC 59

4.1 Integrating DCFNoC into a Manycore Design . . . 60

4.1.1 Tile Architecture . . . 60

4.1.2 Network Interface . . . 60

4.1.3 Modifications to Include TDM and DCFNoC . . . 62

4.1.3.1 End-to-end Flow Control . . . 63

4.1.3.2 Enforcing Deadlock Freedom . . . 63

4.1.3.3 TDM scheduler . . . 64

4.1.3.4 Network Ejection Module . . . 65

4.2 Evaluation Results . . . 66

4.2.1 Experimental Setup . . . 66

4.2.2 Timing Guarantees . . . 66

4.2.3 Performance Guarantees of DCFNoC Manycore System . . . 68

4.2.4 Flexible Bandwidth Allocation in the Manycore . . . 72

4.2.5 Performance Evaluation of Real Workloads in a Manycore . . . 73

4.2.6 Area and Frequency of DCFNoC . . . 74

4.3 Summary . . . 76

5 hp-DCFNoC: High Performance Distributed Dynamic TDM Scheduler based on DCFNoC Theory 77

5.1 A Distributed Dynamic Scheduler Design . . . 78

5.1.1 Scheduler Architecture . . . 78

5.1.2 Notification Phase . . . 80

5.1.3 Data Phase . . . 83

5.1.4 Assignment of TDM slot priorities . . . 84

5.1.5 Rescheduling technique . . . 85

5.2 Evaluation . . . 88

5.2.1 Experimental Setup . . . 88
5.2.2 Theoretical Worst-Case Performance . . . 89
5.2.3 Testing Worst-Case Performance . . . 90
5.2.4 Analyzing the Impact on Applications Behaviour . . . 93
5.2.5 Area Overhead and Frequency . . . 95
5.3 Summary . . . 98

6 A Study on Conflict-Free TDM-based NoC Communications 99
6.1 Families of Solutions . . . 100
6.1.1 Token Propagation-Based Family . . . 102
6.1.1.1 Conflict-Free Intra Domain . . . 106
6.1.1.2 Increasing the Number of Domains . . . 110
6.1.2 DCFNoC Based Family . . . 112
6.1.2.1 DCFNoC Following XY Routing Algorithm . . . 113
6.1.2.2 DCFNoC Following SR Routing Algorithm . . . 114
6.1.3 No-Delay Based Family . . . 115
6.1.3.1 Couples Injection Following XY Routing . . . 116
6.1.3.2 Diagonal Following Custom Routing . . . 118
6.1.3.3 Filling Router Pipeline Stages to Improve TDM Period . . . 121
6.2 Evaluation Results . . . 123
6.2.1 Experimental Setup . . . 123
6.2.2 Performance Evaluation for IOB Router Architectures . . . 124
6.2.3 Performance Evaluation for OB Router Architectures . . . 126
6.2.4 Scalability Analysis . . . 128
6.2.5 Area Overhead And Frequency . . . 131
6.3 Summary . . . 133
6.4 Acknowledgment . . . 133

7 Conclusions 135
7.1 Contributions . . . 136
7.2 Future Directions . . . 137
7.3 Publications . . . 138

A Extending Token Propagation-Based Family 141
A.1 Supporting Specific Topologies with Unidirectional Links . . . 141
A.2 Algorithm To Insert Delays on Overlapped Rings . . . 143
A.3 Extending to 16 Domains . . . 145
A.4 Extending to Larger Networks . . . 146
A.5 Tuning Router Pipeline Stages to Make Latency Hotspots Lighter . . . 149

References 151

List of Figures

1.1 Different MPSoC applications in real-time systems. 2

1.2 Main contributions to safety-critical MPSoC. 5

2.1 Direct topology at left and indirect topology at right. Nodes are represented by circles. 11

2.2 Baseline switch architecture. Centralized switch allocation for virtual channel switches 13

2.3 Stop&Go flow control. 14

2.4 Credit-based flow control. 15

2.5 Virtual cut-through and Wormhole switching techniques. 16

2.6 Virtual channels operation example. 17

2.7 *Deadlock* situation due to busy requested resources (buffers) between messages in a 4×4 mesh. Each colour represents a message. 18

2.8 DOR routing algorithm in a 4×4 mesh. Each colour represents a message. 19

2.9 Segment-based routing with routing restrictions placed at partitions to avoid deadlock situations. 19

2.10 Conflict example between different communication flows at two network links in a 2×2 mesh network using the DOR algorithm. 24

2.11 Design and evaluation flow. 29

2.12 PEAK architecture. 32

2.13 Network interface controller using wormhole network with virtual networks support to provide intra-tile and inter-tile resource connectivity. 32

3.1 2×2 mesh network. End nodes shown as circles. 39

3.2 *CDG* for the 2×2 mesh topology and the DOR routing algorithm. 40

3.3 CDG_l obtained from *CDG*. 41

3.4 CDG_{dl} by destination node. 42

3.5 3×3 mesh topology. 44

3.6 Channel dependency graph of the 3×3 mesh, using DOR. Squares represent links and its name indicates the routers attached to the link. 45

3.7 Layered Channel dependency graph. Potential conflicts are represented by red dashed lines. 45

3.8 Layered Channel dependency graph with added delays (paths shown only from all to 8). Shades represent the TDM-slot of a potential arrangement of messages in the network. 46

3.9 Different bandwidth allocation options. Each box represents an injection slot and each label indicates the end node the slot is assigned to. 50

3.10 Broadcast support using the DOR routing algorithm in a 3 × 3 mesh (paths shown only from 0 to all). Both, injection and ejection end nodes are represented by circles and channels are represented by squares. 51
3.11 DCFNoC router input/output ports connections with output delay registers. 52
3.12 DCFNoC mesh with output delay registers for paths 0 → 3 and 2 → 3. . . 53
3.13 End-to-End latency of DCFNoC vs ILP [1] and PhaseNoC [2]. Y axes starts at 20 to improve the visibility of the comparison. 55
3.14 Scalability of DCFNoC vs ILP [1] and PhaseNoC [2]. 56
3.15 Heterogeneous bandwidth guarantees assignment in a 6 × 6 mesh. 56
3.16 Area overhead. 57
3.17 Maximum attainable clock frequency. 57

4.1 Baseline manycore architecture implementing a configurable number of virtual networks to separate data traffic. 61
4.2 Network interface controller using Wormhole network and modifications to include TDM and support for two DCFNoC networks. 62
4.3 Execution time for benchmarks with different percentage of memory access instructions. 67
4.4 Average memory transaction latency when using only one network. 68
4.5 Execution time when using only one or two networks. 70
4.6 Average memory transaction latency when using only one or two networks. 70
4.7 Scalability of execution time for 4 × 4 and 8 × 8. 71
4.8 Scalability of average memory transaction latency for 4 × 4 and 8 × 8. . . 71
4.9 Execution time using different bandwidth allocations. Application executed alone using a wormhole NoC and using DCFNoC with 1-cycle assigned time slot of 16 and 2-cycle assigned time slots of 17. 73
4.10 Average transaction latency using different bandwidth allocations. 73
4.11 Latency for different benchmarks execution at different core positions $(0, 5, 15)$ in a 4 × 4 mesh system using wormhole and DCFNoC NoCs. . . 74
4.12 Network Interface Area overhead. 75
4.13 Network Interface Maximum attainable clock frequency. 75

5.1 Scheduler phases. Notification phase configures next transmission window. 79
5.2 Scheduler to optimize DCFNoC performance and resource utilization. . . . 79
5.3 First priority rule in case of incompatible paths: Notified route uses the priority slot of the receiver. 81
5.4 Second priority rule in case of incompatible paths: Notified route uses the priority slot of the sender. 82
5.5 Third priority rule in case of incompatible paths: Notified route does not use either the priority slot of the sender nor the receiver's one. 82
5.6 Data phase. At the end of notification phase the slot manager sends information about the slots assigned to each message. 83
5.7 Disjoint paths between node 0 and node 1 in (a). In (b) node 0 and node 15. Paths sharing red resources or destination node are incompatible. . . . 84
5.8 Comparison between common scheduler phases (up) and using rescheduling technique (down). Notification phase configures next transmission window. Data transmission phase is now broken down in $transmX_0$ and $transmX_1$. 86

5.9 Example of notification nodes targeting scheduling slots in common scheduler phases (up) and using rescheduling technique (down). Note that resulting transmission phase when rescheduling is the addition of $transmX_0$ and $transmX_1$. 87

5.10 4 × 4 mesh system with schedulers using two DCFNoC networks. 88

5.11 hp-DCFNoC throughput guarantees for a 4 × 4 and 8 × 8 mesh system. In (a) node 5 is injecting traffic at 10% injection rate while others are injecting at 50%. In (b) nodes $(10, 20, 30, 40, 50$ and $60)$ inject at 50% injection rate while the rest inject at 5%. 90

5.12 Network throughput using baseline scheduler (*BaseSched*) for different ways in a 4 × 4 mesh system. 91

5.13 Network throughput (a) and message latency (b) in a 4 × 4 mesh. 92

5.14 Network throughput (a) and message latency (b) in a 8 × 8 mesh. 92

5.15 Network throughput (a) and message latency (b) comparison versus standard wormhole in a 4 × 4 and 8 × 8 mesh. 93

5.16 Normalized throughput comparison for different scenarios in a 4 × 4 mesh system when using hp-DCFNoC and wormhole NoCs. Application is originated traffic in the farthest node w.r.t to the MC. 95

5.17 Area overhead at each node. 96

5.18 Area overhead for *ImpSched* implementations with different number of ways in a 4 × 4 mesh system. 97

5.19 Maximum attainable clock frequency for all modules of hp-DCFNoC. Wormhole is also shown for comparison purposes. 97

6.1 Router architectures: (a) Two flip-flop router architecture (IOB), (b) One flip-flop router architecture (OB). 101

6.2 Token propagation flow with flooding methodology in a specific time slot for IOB router architecture. Numbers denote the token ID served on a specific NoC resource at the same clock cycle. The token is injected from the bottom right-most node. 103

6.3 Token propagation round-trip time becomes a period of 4 cycles in IOB router architecture. 104

6.4 Token propagation using square shape in a 2D mesh using IOB router architecture. 105

6.5 TDM slot wheel assignment to manage message injection in a 2D mesh with support for 4 domains. Every node is allowed to inject when the slot ID is equal to its node ID. Allowed inject nodes are represented by red circles at TDM slot wheel. Messages are represented by inject node numbers and subindex represents the propagation time in cycles. 107

6.6 Scenario 1, consecutive nodes in the same row injecting to the same X direction. Message injection mechanism in Period 4 solution based on TDM slot wheel assignment using IOB routers. Nodes highlighted in red are the ones injecting. 107

6.7 Scenario 2, corner nodes injecting to the north. 108

6.8 Scenario 3, consecutive nodes in the same row injecting to different X direction. 109

6.9 Scenario 4, consecutive nodes in the same column injecting to the south. . 109

6.10 Flooding propagation in a 2D mesh using IOB routers composing a period of 16 cycles. This is a Period 4 solution which have a natural period of 4 cycles and uses additional delays (12 cycles in every red square) to enlarge the period to 16 cycles. Additional delays are represented by red squares at input ports. Some links are not plotted for representation purposes. . . 110
6.11 Breaking round-trip time limits by additional delays to support more domains. Additional delays are represented by red squares. 111
6.12 TDM slot wheel assignment to manage message injection in a 2D mesh with support for 16 domains. Every node are allowed to inject when the pointed slot ID is equal to its node ID. 111
6.13 Initial DCFNoC approach link order following XY routing algorithm in a 4×4 mesh. Nodes are represented by circles and propagation layers are represented by arrows. Numbers represents every propagation layer thorough the CDG_{dl}. 113
6.14 OSR token propagation phases following XY routing algorithm in a 4×4 mesh using OB routers. Nodes are represented by circles and OSR token propagation phases are represented by arrows. Numbers represents every propagation phase. 114
6.15 CDG_{dl} for the 4×4 mesh topology using OB routers and the XY routing algorithm. 114
6.16 Network-level token propagation, with annotated latency, in the order of the CDG with periodic SR routing (dictating the position of routing restrictions) and single-cycle routers and links. Source [3]. 115
6.17 CDG_l for the 4×4 mesh topology using OB routers and the SR routing algorithm. 116
6.18 Determining the packet travelling time between couples of injectors to avoid conflicts with new injected packets. For clear representation purposes, some packet travelling time are missing. 117
6.19 TDM slot wheel assignment to manage couples message injection in a 2D mesh using IOB routers. Every node are allowed to inject in the pointed slot ID. 118
6.20 Couples injection solution. Numbers at routers represent nodes injection time. Relative difference between injectors by couples. 119
6.21 Couples injection CDG of router 0 and 1. Relative difference between injector nodes $(0, 1)$ and $(4, 5)$ are two hops. 119
6.22 Diagonal approach. Improving TDM period by using more injectors at the same time (columns) as well as custom routing to serialize packets in one point (router 15). Since first phase and second phase are using different network links conflicts are avoided. Destination nodes are only reached at second phase. 119
6.23 Determining the packet travelling time between two columns of injector nodes to avoid conflicts with new injected packets. 121
6.24 Columns injection CDG of third column (worst-case). Relative difference between third column injector nodes $(2, 6, 10, 14)$ and serialization node 15 are four hops. 122
6.25 Relative difference between injector nodes by columns 122

6.26 TDM slot wheel assignment to manage columns message injection in a 2D mesh using IOB routers. Every node are allowed to inject in the pointed slot ID. 122
6.27 Fulfilling pipeline stages of a 4 router ring with a period of 8. 122
6.28 Load latency comparison for IOB router architectures in a 4 × 4 network. 125
6.29 End-to-End latency comparison for IOB router architectures in a 4 × 4 network. 125
6.30 Cost comparison for IOB router architectures in terms of ports and additional delays in a 4 × 4 network. 126
6.31 Load latency comparison for OB router architectures in a 4 × 4 network. . 127
6.32 End-to-End latency comparison for OB router architectures in a 4 × 4 network. 127
6.33 Cost comparison for OB router architectures in terms of ports and additional delays in a 4 × 4 network. 128
6.34 Load latency comparison for IOB router architectures in a 8 × 8 network. 129
6.35 End-to-End latency comparison for IOB router architectures in a 8 × 8 network. 129
6.36 Cost comparison for IOB router architectures in terms of ports and additional delays in a 8 × 8 network. 130
6.37 Area overhead for IOB router architectures in a 4×4 network and DCFNoC in a 8 × 8 network. 132
6.38 Maximum attainable clock frequency for IOB router architectures in a 4 × 4 network and DCFNoC in a 8 × 8 network. 132

A.1 Unidirectional ring of 4 routers with a period of 8 cycles using IOB routers.142
A.2 Combining multiple unidirectional rings of 4 preserving a relative latency of 8 cycles in every converging point using IOB routers. 142
A.3 Optimal number of domains equal to 8 in a 4 × 4 mesh using IOB routers by only combining unidirectional rings of 4 routers. The token is injected from the bottom right-most node. 142
A.4 Breaking unidirectional rings by additional delays to support more domains using IOB routers. Additional delays are represented by red rectangles. 143
A.5 Period 8 solution in a 4 × 4 mesh by combining unidirectional rings of 4 IOB routers and adding delays (8 cycles in every red square) to enlarge the period to 16 cycles. Algorithm A.1 is applied to insert delays. Additional delays are represented by red squares. 145
A.6 TDM slot wheel assignment to manage message injection in a 4 × 4 mesh by combining unidirectional rings of 4 IOB routers with support for 16 domains. 145
A.7 Unidirectional ring of 8 routers with a period of 16 cycles. 146
A.8 Period 16 topology in a 4 × 4 mesh by only combining unidirectional rings of 8 routers using IOB routers. This is a delay-free solution. 146
A.9 TDM slot wheel assignment to manage message injection in a 4 × 4 mesh by combining unidirectional rings of 8 routers with support for 16 domains.146

A.10 Period 32 topology in a 8 × 8 mesh by combining unidirectional rings of 16 routers and adding delays (32 cycles in every red square) to enlarge the period to 64 cycles. Implemented delays are 32 cycles at every red point. This is an extension of Period 16 for a 4 × 4 mesh 147
A.11 Period 64 topology in a 8 × 8 mesh by combining unidirectional rings of 32 routers. This is a delay-free solution. 147
A.12 Period 16 topology in a 8 × 8 mesh by combining unidirectional rings of 16 routers and adding delays enlarge the period to 64 cycles. Implemented delays are 48 cycles at every red point. 148
A.13 Period between two consecutive routers in Period 4 topology (a). In order to spread the added delays, we move them from some input ports to all output ports, and hence adding one pipeline stage (b). 149
A.14 Gradually moving the added delays by adding two pipeline stages (a). By adding six pipelines stages every hop have the same length (b). 150

List of Tables

3.1 Summary of Notation 38
3.2 A comparison of schedule length and network latency. 54

5.1 Different scenarios evaluated. We have modeled three main different scenarios for application traffic with four different levels of network congestion. 94

6.1 Summary of Concepts 102