

Efficient Communication in Heterogeneous Architectures

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

Doctor of Philosophy

BY

NARAYANA SRI HARSHA GADE



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI– 110020

AUGUST, 2020

Acknowledgements

This thesis has been made possible through the support and assistance of several individuals in one way or the other throughout my amazing journey as a graduate student. I extend my heartfelt thanks to each and everyone of them.

First and foremost, I express my sincere gratitude and am grateful to my advisor Dr. Sujay Deb for providing me with the opportunity and his guidance through all these years. His valuable suggestions, technical inputs and continuous support through the years have encouraged me to learn and expand my knowledge and his insights during various stages of my research is what has made this thesis possible today. Besides his guidance in my work, he has always encouraged and motivated me to push beyond limits, create opportunities and make my own way in my career. I am forever indebted to him both for his professional and personal support through all these years. I also extend my gratitude to Dr. Shobha Sundar Ram (Indraprastha Institute of Information Technology Delhi) and Dr. Amlan Ganguly (Rochester Institute of Technology) for their guidance and support in completion of this thesis.

This thesis and my PhD itself would not have seen culmination if not for my family. They have given me the freedom to choose my career and have had faith in me, no matter what. My mother has always had been my constant support throughout all my endeavours and I sincerely thank her from the bottom my heart for her confidence in me.

No graduate and PhD journey is ever complete without some amazing friends. I can honestly say that I have been the luckiest in meeting two best friends in Anupriya and Alvika during my time as PhD scholar. They have always been there to support me through my work, share my joy in success and cheer me up and motivate me when things have not gone well. Anupriya and me have shared the same lab space and it is the continuous banter with her that had made my time at lab such a joyous ride. She has been there for everything from work

discussions to impromptu outings to many many travel plans and in her I have a great friend, motivator and travel partner. Alvika had been a partner in crime through endless cups of coffee, parties, late night working and pulling all nighters to wrap up papers. Through her, I have learned one of the greatest qualities to possess, dedication to one's work. I have had many cherished memories with both of them during my time at IIIT Delhi and it is an understatement to say I have missed them during the last days of the PhD journey.

I have also had the chance to work with great colleagues during my PhD like Hemanta Mondal, Wazir Singh, Mitali Sinha and Sidhartha Rout. I thank them for their time in many technical discussions and their valuable inputs on my work. I would also like to thank the undergraduate and graduate who have worked with me during the years. I also thank Dr. Sudeep Pasricha from Colorado State University for providing me with an opportunity to work with him as a research fellow. Last but not least, I would like to express my gratitude to administrative staff at IIIT Delhi, especially Priti Patel for all their help in logistics and office related matters.

Certificate

This is to certify that the thesis titled “Efficient Communication in Heterogeneous Architectures” being submitted by Narayana Sri Harsha Gade to INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI, for the award of the DOCTOR OF PHILOSOPHY, is an original research work carried out under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

Dr. Sujay Deb

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

Abstract

Advanced CMOS manufacturing technologies and limitations of power scaling have ushered processing chip architectures into the era of multi-core and many core systems. Simultaneously, there has been an emergence of compute and memory intensive applications like machine learning, cloud computing, artificial intelligence, etc. along with scientific applications, that require ever increasing performance demands. With diminishing returns from complex superscalar architectures, heterogeneous many core architectures have gained significant interest to meet the performance demands of these wide range of applications. They integrate processing cores of different architectures like x86, Field Programmable Gate Arrays (FPGAs), Graphic Processing Units (GPUs), application specific architectures, etc. in a single system. They have several heterogeneous cores working together in parallel on different application kernels towards finding good balance in meeting each kernel's respective processing requirements and achieving high performance. Especially, single chip and multi-chip Heterogeneous System Architectures (HSA), that combine GPUs with x86 processors have become wide spread for meeting performance demands of highly parallel general purpose applications.

Integration of heterogeneous architectures in a single system, while efficient and becoming wide spread, still poses several challenges for cohesive design and ubiquitous deployment. These challenges range from (i) circuit level for low power integration to (ii) system architectural challenges like shared resources, resource allocation per heterogeneous architecture, etc. for power and performance efficiency to (iii) software and application level challenges for seamless use of heterogeneous resources for achieving high performance. One such major challenge for heterogeneous architectures and the focus of this thesis is the communication across heterogeneous components and the shared system resources of on-chip interconnection network and memory/cache hierarchy. In the context of this thesis, communication in heterogeneous architectures is considered to be broadly comprised of data and coherence control communication.

Communication and interconnection topologies are one of the major performance and power bottlenecks

for architectures that integrate many cores, whether homogeneous or heterogeneous. As core counts increase, processing architectures shift from computation to communication centric systems and data and coherence communication become the significant part of chip performance and power. Heterogeneous architectures pose additional communication challenges due to the differences in computing modalities of each architecture. The major of these being the highly diverse ways in which each of these architectures request for data and communicate with other cores/memory. These different and potentially conflicting traffics on the shared interconnection backbone lead to interference and suboptimal performance among the integrated architectures. Other challenges arise from presence of shared data between different computing architectures (coherence), operational heterogeneity, inefficient memory sharing, application kernel partitioning and communication policies, etc.

To overcome the major challenge of communication diversity in heterogeneous architectures, we propose HyWin interconnection, built around customized networks and wireless links. HyWin decouples different architectures from one another to minimize interference among them and improve performance across all architectures. Besides improving communication performance, HyWin reduces overheads in presence of operational heterogeneity and reduces design complexity when extending to different heterogeneous architectures. It is robust and easily scales from heterogeneous single chip processors to multi-die and multi-chip systems, irrespective of the integrated architectures. Utilizing the HyWin architecture, we provide scalable and efficient cache coherence through SpyDir and WiSH protocols. SpyDir is a hierarchical hybrid coherence, that combines local directory with global snoopy protocol to minimize coherence storage and energy overheads, while providing necessary performance at any system size. WiSH further reduces coherence overheads by incorporating application sharing characteristics through logical share aware segmentation of caches. On-chip wireless links form the backbone for both HyWin interconnection and SpyDir coherence and are pivotal for realizing high performance and energy efficiency. Towards this end, we perform comprehensive analysis of on-chip wireless channel and design robust and high bandwidth wireless links for use in proposed approaches. HyWin and WiSH addresses the major challenges of communication diversity on shared network and coherence communication in heterogeneous architectures and provides a heterogeneity agnostic interconnection design and coherence protocol that leverages it to provide efficient communication in heterogeneous systems.

List of publications

Journals

1. Sri Harsha Gade, M. Meraj Ahmed, Sujay Deb and Amlan Ganguly, "Energy Efficient Chip-to-Chip Interconnection for Heterogeneous Architectures," in *ACM Trans. Des. Autom. Electron. Syst.* 24, 5, Article 55 (July 2019), 27 pages. (DOI: 10.1145/3340109)
2. Sri Harsha Gade, Shobha Sundar Ram and Sujay Deb, "Millimeter Wave Wireless Interconnects in Deep Submicron Chips: Challenges and Opportunities," in *Integration*, vol. 64, pp. 127-136, 2019. (DOI: 10.1016/j.vlsi.2018.09.004)
3. Sri Harsha Gade and Sujay Deb, "HyWin: Hybrid Wireless NoC with Sandboxed Sub-Networks for CPU/GPU Architectures," in *IEEE Transactions on Computers*, vol. 66, no. 7, pp. 1145-1158, 1 July 2017. (DOI: 10.1109/TC.2016.2643668)
4. Sri Harsha Gade and Sujay Deb, "WiSH: Wireless Enabled, Share Aware Hybrid Cache Coherence for Many Core Architectures," in *ACM Transactions on Design Automation of Electronic Systems*. (Under review)

Conferences

1. Sri Harsha Gade, Sidhartha Sankar Rout, Ravi Kashyap and Sujay Deb, "Reliability Analysis of On-Chip Wireless Links for Many Core WNoCs," 2018 Conference on Design of Circuits and Integrated Systems (DCIS), Lyon, France, 2018, pp. 1-6. (DOI: 10.1109/DCIS.2018.8681475)
2. Sri Harsha Gade, Sidhartha Sankar Rout and Sujay Deb, "On-Chip Wireless Channel Propagation: Impact of Antenna Directionality and Placement on Channel Performance," 2018 Twelfth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), Turin, 2018, pp. 1-8. (DOI: 10.1109/NOCS.2018.8512173)
3. Sri Harsha Gade, Hemanta Kumar Mondal and Sujay Deb, "High Bandwidth Off-Chip Memory Access Through Hybrid Switching and Inter-Chip Wireless Links," 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Hong Kong, 2018, pp. 100-105. (DOI: 10.1109/ISVLSI.2018.00028)
4. Sri Harsha Gade, Mitali Sinha, Sidhartha Sankar Rout and Sujay Deb, "Enabling Reliable High Throughput On-chip Wireless Communication for Many Core Architectures," 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Hong Kong, 2018, pp. 591-596. (DOI: 10.1109/ISVLSI.2018.00113)

5. Sri Harsha Gade, Sidhartha Sankar Rout, Mitali Sinha, Hemanta Kumar Mondal, Wazir Singh and Sujay Deb, "A Utilization Aware Robust Channel Access Mechanism for Wireless NoCs," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, 2018, pp. 1-5. (DOI: 10.1109/ISCAS.2018.8351698)
6. Shashwat Kaushik, Muni Agrawal, Hemanta Kumar Mondal, Sri Harsha Gade and Sujay Deb, "Path loss-aware adaptive transmission power control scheme for energy-efficient wireless NoC," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, 2017, pp. 132-135. (DOI: 10.1109/MWSCAS.2017.8052878)
7. Sri Harsha Gade, Sakshi Garg and Sujay Deb, "OFDM Based High Data Rate, Fading Resilient Transceiver for Wireless Networks-on-Chip," 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Bochum, 2017, pp. 483-488. (DOI: 10.1109/ISVLSI.2017.90)
8. Hemanta Kumar Mondal, Shashwat Kaushik, Sri Harsha Gade and Sujay Deb, "Energy-Efficient Transceiver for Wireless NoC," 2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems (VLSID), Hyderabad, 2017, pp. 87-92. (DOI: 10.1109/VLSID.2017.42)
9. Sri Harsha Gade and Sujay Deb. 2015. Achievable Performance Enhancements with mm-Wave Wireless Interconnects in NoC. In Proceedings of the 9th International Symposium on Networks-on-Chip (NOCS '15). ACM, New York, NY, USA, Article 29, 2 pages. (DOI: 10.1145/2786572.2786584)

Other Publications

1. Hemanta Kumar Mondal, Sri Harsha Gade, Shashwat Kaushik and Sujay Deb, "Adaptive Multi-Voltage Scaling with Utilization Prediction for Energy-Efficient Wireless NoC," in IEEE Transactions on Sustainable Computing, vol. 2, no. 4, pp. 382-395, 1 Oct.-Dec. 2017. (DOI: 10.1109/TSUSC.2017.2742219)
2. Hemanta Kumar Mondal, Sri Harsha Gade, Raghav Kishore and Sujay Deb, "P2NoC: Power- and Performance-aware NoC Architectures for Sustainable Computing," in Sustainable Computing: Informatics and Systems, vol. 16, pp. 25-37, 2017. (DOI: 10.1016/j.suscom.2017.08.005)
3. Hemanta Kumar Mondal, Sri Harsha Gade, Md Shahriar Shamim, Sujay Deb and Amlan Ganguly, "Interference-Aware Wireless Network-on-Chip Architecture using Directional Antennas," in IEEE Transactions on Multi-Scale Computing Systems, vol. 3, no. 3, pp. 193-205, 1 July-Sept. 2017. (DOI: 10.1109/TMSCS.2016.2595527)
4. Mitali Sinha, Sri Harsha Gade, Pramit Bhattacharyya and Sujay Deb, "Design Space Optimization of Shared Memory Architecture in Accelerator-rich Systems," in ACM Transactions on Design Automation of Electronic Systems. (Under Review)
5. Mitali Sinha, Sri Harsha Gade, Wazir Singh and Sujay Deb, "Data-flow Aware CNN Accelerator with Hybrid Wireless Interconnection," 2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP), Milan, 2018, pp. 1-4. (DOI: 10.1109/ASAP.2018.8445126)
6. Sidhartha Sankar Rout, Hemanta Kumar Mondal, Rohan Juneja, Sri Harsha Gade and Sujay Deb, "Dynamic NoC platform for varied application needs," 2018 19th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, 2018, pp. 232-237. (DOI: 10.1109/ISQED.2018.8357293)
7. Mitali Sinha, Sidhartha Sankar Rout, Sri Harsha Gade and Sujay Deb, "Near Threshold Last Level Cache for Energy Efficient Embedded Applications," 2018 Ninth International Green and Sustainable Computing Conference (IGSC), Pittsburgh, PA, USA, 2018, pp. 1-6. (DOI: 10.1109/IGCC.2018.8752134)

8. Shrestha Bansal, Hemanta Kumar Mondal, Sri Harsha Gade and Sujay Deb, "Energy Efficient NoC Router for High Throughput Applications in Many-Core GPUs," 2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS), Bhopal, 2017, pp. 50-55. (DOI: 10.1109/iNIS.2017.20)
9. Hemanta Kumar Mondal, Sri Harsha Gade, Raghav Kishore and Sujay Deb, "Adaptive multi-voltage scaling in wireless NoC for high performance low power applications," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2016, pp. 1315-1320.
10. Hemanta Kumar Mondal, Sri Harsha Gade, Raghav Kishore, Shashwat Kaushik and Sujay Deb, "Power efficient router architecture for wireless Network-on-Chip," 2016 17th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, 2016, pp. 227-233. (DOI: 10.1109/ISQED.2016.7479205)
11. Sri Harsha Gade, Praveen Kumar and Sujay Deb, "A Pre-RTL floorplanner tool for automated CMP design space exploration with thermal awareness," 2016 20th International Symposium on VLSI Design and Test (VDATE), Guwahati, 2016, pp. 1-6. (DOI: 10.1109/ISVDATE.2016.8064876)
12. Hemanta Kumar Mondal, Sri Harsha Gade, Raghav Kishore and Sujay Deb, "Power- and performance-aware fine-grained reconfigurable router architecture for NoC," 2015 Sixth International Green and Sustainable Computing Conference (IGSC), Las Vegas, NV, 2015, pp. 1-6. (DOI: 10.1109/IGSC.2015.7393682)

Contents

1	Introduction	1
1.1	Heterogeneous Architectures	1
1.2	Challenges in Heterogeneous Architectures	2
1.3	Communication Challenges in HSA	3
1.3.1	Data communication challenges in heterogeneous architectures	4
1.3.2	Cache coherence communication challenges	5
1.4	Research Contributions	6
1.4.1	Design of efficient interconnection for heterogeneous architectures	6
1.4.2	Design of scalable and efficient coherence protocol	7
1.4.3	Wireless channel modelling and link design	7
1.5	Organization of Thesis	8
2	Efficient Interconnection in Single Chip HSA	9
2.1	Communication Challenges in Single Chip HSA	9
2.2	Related Work	12
2.3	HyWin NoC Topology	13
2.3.1	Hybrid WNoC with Sandboxed Networks	13

2.3.2	CPU/GPU Heterogeneous System Architecture	15
2.3.3	HyWin Topology for CPU/GPU HSA	16
2.3.4	WIs as Interconnection Backbone	19
2.3.5	Routing Algorithm	20
2.4	Experimental Results	22
2.4.1	Experimental Setup	22
2.4.2	Performance Evaluation	25
2.4.3	Energy Savings	29
2.4.4	Scalability	30
2.4.5	Comparison with Existing Works	32
2.4.6	Overheads	32
2.5	Conclusions and Future Work	33
3	Hybrid Wireless Interconnection for Multi Chip HSA	35
3.1	Communication Challenges in Multichip Heterogeneous Systems	35
3.2	Related Work	38
3.2.1	State-of-the-Art	38
3.2.2	Emerging Interconnects	39
3.3	Wireless Interconnection Architecture for Heterogeneous Systems	40
3.3.1	CPU/GPU Heterogeneous System Architecture	40
3.3.2	Inter-Chip Wireless Network	43
3.4	Inter-Chip Wireless Infrastructure	45
3.4.1	Omnidirectional Antenna	46

3.4.2	Directional Antenna	46
3.4.3	Transceiver Circuitry	46
3.5	Wireless Communication Protocol	48
3.5.1	Medium Access Control (MAC)	48
3.5.2	Flow Control and Routing	49
3.6	Experimental Evaluation	50
3.6.1	Experimental Setup	50
3.6.2	Hybrid Wireless Interconnection	53
3.6.3	Network Energy	53
3.6.4	Performance Evaluation	57
3.6.5	Overhead Analysis	62
3.7	Extension to Any Heterogeneous Architecture	63
3.8	Conclusions and Future Work	65
4	Wireless Enabled Hybrid Coherence for Many Core Architectures	67
4.1	Cache Coherence Challenges	67
4.2	WiSH Coherence	70
4.2.1	Chiplet or Clustered Many Core Architecture with HyWin Interconnection	70
4.2.2	Hybrid Snoopy over Directory (SpyDir) Protocol	72
4.2.3	Share Aware Cache Segmentation	75
4.2.4	SpyDir Coherence Requests and Status Updates to Cache Lines	79
4.3	Experimental Results	81
4.3.1	Experimental Setup	81

4.3.2	SpyDir Protocol Performance: Monolithic Architectures	84
4.3.3	SpyDir Protocol Performance: Chiplet Architectures	87
4.3.4	Evaluation of WiSH Coherence	89
4.3.5	Overheads	92
4.4	Summary of Existing Works and Comparison with WiSH Coherence	93
4.4.1	Comparison with Existing Works	96
4.5	Conclusions and Future Work	97
5	Wireless Channel Modelling and Link Design	99
5.1	Wireless Channel Modelling	99
5.2	Wireless Channel Estimation and Link Performance: Challenges	102
5.2.1	Intra Chip Geometries	103
5.2.2	Near Field Effects	103
5.2.3	Multipath Propagation	103
5.2.4	Channel Estimation	104
5.3	On-Chip Model and Simulation Setup for Channel Estimation	104
5.4	Impact of Chip Layers on Propagation Paths	107
5.4.1	Time and Frequency Domain Analysis	107
5.4.2	Propagation Components	108
5.5	Implementation of On-Chip Antennas	109
5.5.1	Simulation Methodology	111
5.6	Analysis of Intra Chip Wireless Channel	112
5.6.1	Analysis of Antenna Efficiency	112

5.6.2	On-Chip Wireless Channel	114
5.7	Impact of Antenna Placement and Directionality	118
5.7.1	Time Domain Analysis	118
5.7.2	Propagation Delay	120
5.7.3	Delay Spread	120
5.7.4	Path Loss	121
5.8	On-Chip Wireless Link Design Guidelines	122
5.9	Impact on Wireless Interconnect Performance	123
5.9.1	Interconnect Latency	124
5.9.2	Energy Consumption	125
5.9.3	Network Performance	125
5.10	Impact on WNoC	126
5.11	Design of Wireless Transceiver	127
5.11.1	OFDM Transceiver for WNoCs	128
5.11.2	Implementation of OFDM Transceiver	129
5.11.3	Analysis of OFDM Transceiver	131
5.11.4	Evaluation of Network Performance	133
5.12	Robust Wireless Channel Access Mechanism	134
5.12.1	Comparator based Controller	135
5.12.2	Transceiver Power Gating Control	137
5.12.3	Performance of Comparator based Controller	138
5.13	Concurrent Transmission Enabled Wireless Network Design	140

5.13.1	Concurrent Transmissions using Sub-Channel Grouping	140
5.13.2	Data Reception and Decoding	141
5.13.3	OFDM Transceiver for Concurrent Transmission	142
5.13.4	WNoC Performance with Concurrent Transmissions	143
5.14	Reliability Analysis of Wireless Links	145
5.14.1	On-Chip Wireless Link Setup for Reliability Analysis	145
5.14.2	Reliability Analysis	146
5.14.3	Impact on WNoC	149
5.15	Conclusions and Future Work	149

6	Conclusions and Future Directions	151
----------	--	------------

List of Tables

2.1	Details of CPU/GPU Heterogeneous System Architecture	22
2.2	Details of HyWin Topology for CPU/GPU HSA	23
2.3	Benchmark Characteristics for CPU and GPU Applications	24
2.4	Sub-network Topologies at Different Subsystem Sizes	31
2.5	Comparison of HyWin Topology with Existing Works	31
3.1	Dimensions of PLPA for Directional Wireless Links	46
3.2	Details of CPU/GPU Heterogeneous Systems	51
3.3	Details of Intra and Inter Chip Network	52
3.4	Total Bandwidth of the Proposed Inter-Chip Wireless Interconnection	52
3.5	CPU and GPU Applications Evaluated	53
3.6	Area and Power Overheads of Inter-Chip Wireless Interconnection	62
4.1	SpyDir Protocol States in WiSH	77
4.2	System Setup for Evaluation of WiSH Coherence	81
4.3	Details of Network Topology for WiSH	82
4.4	Benchmarks Characteristics for WiSH Coherence Evaluation	83
4.5	Area and Power Overheads of WiSH Coherence	93

4.6	Summary of WiSH Coherence along with Existing Works	96
5.1	Summary of Existing and Proposed Works for On-Chip Wireless Link Analysis	101
5.2	Material Properties and Dimensions of the On-Chip Propagation Environment Model	105
5.3	Comparison between Wired, Free Space Wireless and Intra Chip Wireless Links	124
5.4	Performance of Proposed and Existing Transceivers for WNoCs	132

List of Figures

1.1	(a) Heterogeneous Architecture with big, little cores, GPU and accelerators, (b) Distributed Heterogeneous Architecture and (c) Heterogeneous Architecture with CPU, GPU and Reconfigurable Cores	2
2.1	Impact on application and network performance due to heterogeneous elements on chip	10
2.2	Illustrative HyWin Topology with Heterogeneous Sub-networks Interconnected by mm-wave WIs	14
2.3	CPU/GPU HSA with Shared Memory Implemented in Our Work	16
2.4	HyWin Topology for the Implemented CPU/GPU HSA	17
2.5	Baseline Mesh Topology for Implemented HSA	24
2.6	Improvement in CPU and GPU runtime using HyWin topology over baseline mesh and WNoC for different application scenarios	26
2.7	Memory access latency in cycles using Mesh, HyWin and WNoC topologies over baseline mesh for different application scenarios	27
2.8	Improvement in peak network bandwidth using HyWin topology over baseline mesh and WNoC for different application scenarios	28
2.9	Savings in average packet energy using HyWin topology over baseline mesh for different application scenarios	29
2.10	Scalability Analysis of HyWin Topology	30

3.1	CPU/GPU heterogeneous multichip system and intra-chip topologies of different processing chips	41
3.2	Baseline Inter-Chip Interconnection for Heterogeneous Multichip Systems with Buses and PCIe Links	42
3.3	Chip-to-Chip Wireless Interconnection with Directional and Omnidirectional Wireless Links for Heterogeneous Multichip Systems	43
3.4	Omnidirectional and Directional Antennas in the Proposed Hybrid Wireless Inter-Chip Interconnection	45
3.5	Transceiver Circuitry of the Proposed Hybrid Wireless Inter-Chip Interconnection	47
3.6	Flit Structure for Medium Access Control and Flow Control for Wireless Transmission	48
3.7	Comparison between hybrid wireless interconnection and omnidirectional only wireless network	53
3.8	Average packet energy saving achieved by wireless inter chip interconnection against traditional wired network on system with two CPU and two GPU chips	54
3.9	Average packet energy saving achieved by wireless inter chip interconnection against traditional wired network on system with four CPU and four GPU chips	55
3.10	Breakdown of network energy consumption on baseline and wireless inter-chip interconnection for different application scenarios	56
3.11	Reduction in average packet latency using wireless inter chip interconnection against traditional wired network on system with two CPU and two GPU chips	57
3.12	Reduction in average packet latency using wireless inter chip interconnection against traditional wired network on system with four CPU and four GPU chips	58
3.13	Breakdown of average packet latency on baseline and wireless inter-chip interconnection for different application scenarios	59
3.14	Application runtime speedup using wireless inter chip interconnection against traditional wired network on system with two CPU and two GPU chips	60

3.15	Application runtime speedup using wireless inter chip interconnection against traditional wired network on system with four CPU and four GPU chips	61
4.1	Many Core System Architecture with Core Clusters or Chiplets Interconnected by Wireless Links	71
4.2	Abstract Representation of Hybrid SpyDir Coherence Protocol	73
4.3	Incorporation of Memory Sharing Patterns through Logical Segmentation of Caches in WiSH .	76
4.4	State Machine Diagram for Protocol Transitions and Sharing State Field Updates in WiSH Coherence	77
4.5	SS Field Updates, Directory and Snoopy Coherence Requests for Load Operation in WiSH . .	79
4.6	SS Field Updates, Directory and Snoopy Coherence Requests for Store Operation in WiSH . .	80
4.7	OFDM based Transceiver for Inter-Cluster Communication	82
4.8	Average Coherence Traffic Reduction provided by SpyDir Protocol against DoS and Dir2 Protocols for Different Benchmarks	85
4.9	Average Speedup in Application Runtime provided by SpyDir Protocol against DoS and Dir2 Protocols for Different Benchmarks	86
4.10	Average Packet Energy Saving provided by SpyDir Protocol against DoS and Dir2 Protocols for Different Benchmarks	86
4.11	Reduction in Coherence Traffic through SpyDir as compared to Two-level Directory Protocol .	87
4.12	Application Runtime Speedup through SpyDir as compared to Two-level Directory Protocol .	88
4.13	Reduction in Network Energy through SpyDir as compared to Two-level Directory Protocol .	88
4.14	Average Reduction in Coherence Traffic provided by WiSH against DoS and Dir2 with Different Cluster Sizes for Different Benchmarks	89
4.15	Application Runtime Speedup provided by WiSH against DoS and Dir2 with Different Cluster Sizes for Different Benchmarks	91

4.16	Packet Energy Saving provided by WiSH against DoS and Dir2 with Different Cluster Sizes for Different Benchmarks	92
4.17	Characteristics of Ideal Cache Coherence and Existing Optimizations to Hardware Coherence Schemes	94
5.1	3D on-chip environment with a silicon substrate, copper interconnects and silicon-di-oxide for wireless channel estimation	105
5.2	Electromagnetic wave propagation in a two-dimensional on-chip environment with a silicon substrate, copper interconnect layer and dielectric layer. The propagation includes following paths: (1) direct free space wave signal from transmitter to receiver; (2) signal reflected off the free space and dielectric layer interface; (3) surface wave propagating through dielectric layer and (4) reflected waves from the metal interconnect layers	106
5.3	The transmitter and receiver are 10mm apart in a two-dimensional on-chip environment with different types of interconnect layer configurations. The figure shows (a) the time domain response (b) frequency domain magnitude response and (c) phase response of the received signal	106
5.4	(a) Magnitude and (b) Phase response of different received signal components at 10mm distance of from the transmitter in on-chip environment	108
5.5	On-Chip Antenna Structures Simulated	110
5.6	Return Loss in Three Dimensional On-Chip Environment for Different Configurations	113
5.7	Variation of Path Loss with Distance from Transmitting Antenna	114
5.8	Variation of Delay Spread with Distance from Transmitting Antenna	115
5.9	Variation of Propagation Delay with Transmission Distance	116
5.10	Top View of the Transmitting and Receiver Antenna Placements	117
5.11	Time Domain Representation of the Received Signal for Folded Dipole Antenna at (a) three locations diagonally across the chip from nearest to the farthest distance and (b) three locations orthogonal to the antenna	118

5.12	Time Domain Representation of the Received Signal for Planar Log Periodic Antenna at (a) three locations diagonally across the chip from nearest to the farthest distance and (b) three locations orthogonal to the antenna	119
5.13	Variation of First Peak Delay of the Received Signal with Different Placements for (a) Folded Dipole and (b) Planar Log Periodic Antenna	119
5.14	Variation of Delay Spread in the Received Signal with Different Placements for (a) Folded Dipole Antenna and (b) Planar Log Periodic Antenna	120
5.15	Variation of Path Loss with Different Placements for (a) Folded Dipole Antenna and (b) Planar Log Periodic Antenna	121
5.16	A Typical Communication Path using Wireless Interfaces with all Components in NoC	124
5.17	Comparison of Network Performance with On-Chip and Free Space Wireless Communication Models	126
5.18	OFDM based Transceiver for WNoC	130
5.19	Variation of BER for different OFDM Specifications	132
5.20	Performance and Energy Improvements using Proposed OFDM Scheme for Different Benchmarks	133
5.21	Comparator based Controller for Channel Access and Power Gating in WNoCs	135
5.22	Format of Packet Transmitted over Wireless Link for Comparator based Channel Allocation .	136
5.23	(a) Improvement in Wireless Channel Utilization and Token Passes (b) Average Number of Token Passes across all Transceivers for Different Packet Sizes and (c) Decrease in Average Packet Latency and Energy Consumption using Comparator based Controller	138
5.24	(a) Pattern Matching Circuit to Detect Packet Receiver (b) Data Format Transmitted over Sub-Channel Group	141
5.25	OFDM based Transceiver for Concurrent Transmissions in WNoC	142
5.26	(Improvement in (a) Memory Access Latency and (b) Network Bandwidth using WNoC with Concurrent Transmissions over Wired Mesh	143

5.27	(Improvement in (a) Network Energy and (b) Application Runtime using WNoC with Concurrent Transmissions over Wired Mesh	144
5.28	End-to-End Implementation of On-Chip Wireless Link for Reliability Analysis	145
5.29	Comparison of Bit Error Rate of Free Space and On-Chip Wireless Channels for Different Modulation Schemes	146
5.30	Comparison of Power Spectral Density of Free Space and On-Chip Wireless Channels for Different Modulation Schemes	148
5.31	Increase in Average Packet Energy due to higher Transmission Power Requirement in On-Chip Wireless Channel	148

Chapter 1

Introduction

Multi-core and many core processors have emerged as the alternative to the limitations of Dennard scaling[1] for improving the performance. They help realize high performance computing systems required for emerging applications like IoT, cloud computing, machine learning and artificial intelligence, etc. along with scientific research applications. But, with diminishing returns from increasing number of complex superscalar processors, heterogeneous architectures have gained significant interest to meet the ever increasing performance demands of these wide range of applications.

1.1 Heterogeneous Architectures

Heterogeneous systems integrate processing cores of different architectures like x86, Field Programmable Gate Arrays (FPGAs), Graphic Processing Units (GPUs), application specific architectures, etc. in a single system. They have several heterogeneous cores working together in parallel on different application kernels towards finding good balance in meeting each kernel's respective processing requirements and achieving high performance. Besides architectural heterogeneity, most current day computing processors maintain operational heterogeneity for performance and power efficiency. Figure 1.1 shows different variations of such heterogeneous architectures, which combine architectures in both integrated and distributed manners[2, 3, 4]. Especially, single chip and multi-chip Heterogeneous System Architectures (HSA), that combine GPUs with x86 processors have become wide spread for meeting performance demands of highly parallel general purpose applications. They find a good balance between single thread performance and high throughput applications. State-of-the-art HSA like AMD APU's[5, 6, 7], Intel Core architectures (Sandybridge and later)[8, 9, 10, 11] and nVIDIA's DGX-1 System[12] have shown considerable performance improvements over homogeneous architectures and

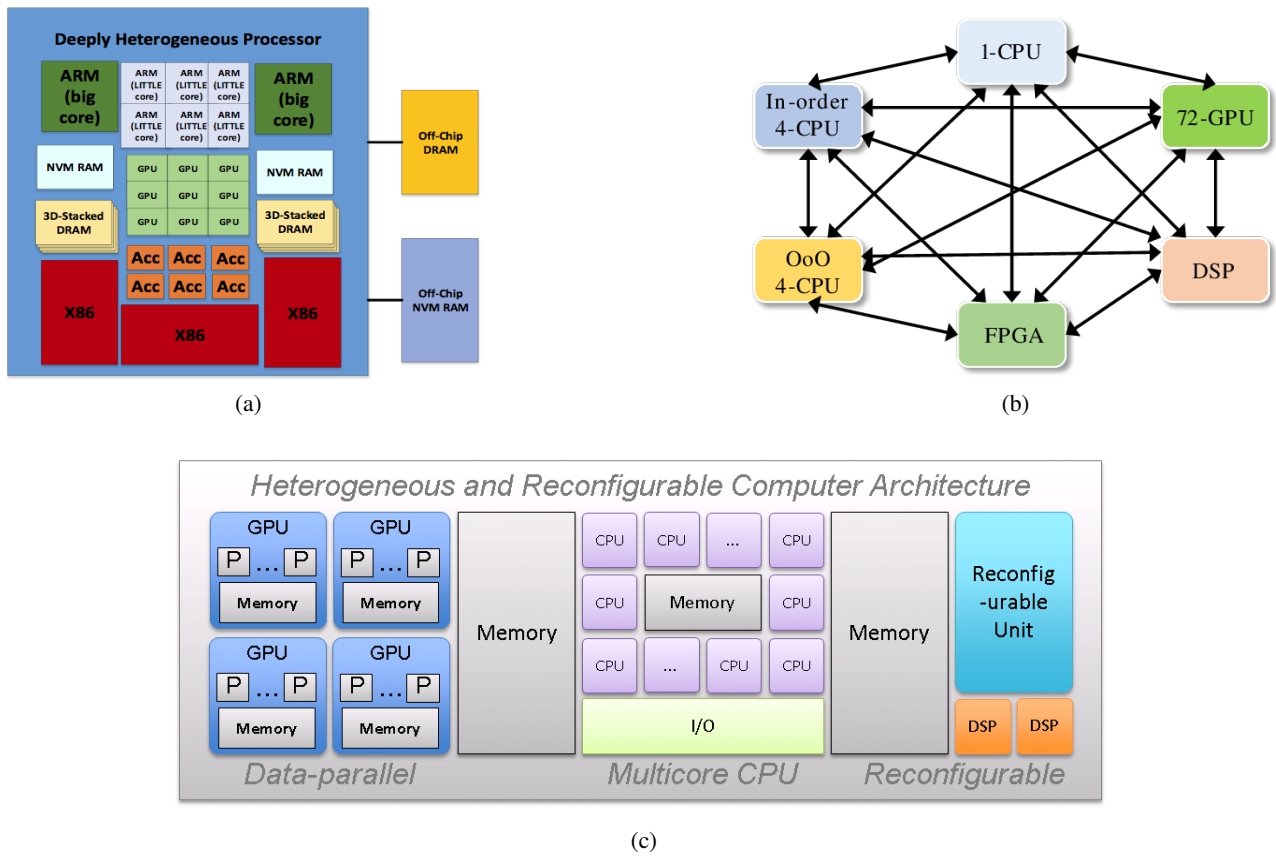


Figure 1.1: (a) Heterogeneous Architecture with big.little cores, GPU and accelerators, (b) Distributed Heterogeneous Architecture and (c) Heterogeneous Architecture with CPU, GPU and Reconfigurable Cores

will become more common in the future to meet wide application requirements in an efficient manner.

1.2 Challenges in Heterogeneous Architectures

Integration of heterogeneous architectures in a single system, while efficient and becoming wide spread, still poses several challenges for cohesive design and ubiquitous deployment. These challenges range from circuit level for low power integration to system architectural challenges like shared resources, resource allocation per heterogeneous architecture, etc. for power and performance efficiency to software and application level challenges for seamless use of heterogeneous resources for achieving high performance. Below is a list of potential challenges in designing and implementing heterogeneous architectures:

- **Instruction Set Architecture (ISA)**: Processing elements in heterogeneous architectures have different ISAs, that interpret applications and memory in different ways, leading to cross-architecture incompatibility and impact application performance adversely.

- **Low level implementation:** Compute elements, even of same ISA can have different micro-architectural and circuit level implementation, that lead to various levels of performance and power consumption. Micro-architectural decisions for heterogeneous compute elements are all the more complex.
- **Memory hierarchy:** Heterogeneous compute elements may work efficiently with different cache hierarchies, coherency protocols, and memory access patterns (uniform or non-uniform), arbitrary data transfers, burstiness and endianness, etc. Presence of shared memories across heterogeneous architectures can lead to interference among applications running on any or all of the architectures.
- **Interconnection:** Compute elements in heterogeneous systems may have differing types of interconnect like basic memory/bus or dedicated network interfaces, memory controllers, Direct memory access (DMA), link or data sizes, etc. Furthermore, certain portions of a heterogeneous system may be cache-coherent, whereas others may require explicit software-involvement for maintaining coherency.
- **Data partitioning and application programming:** Partitioning data on heterogeneous architectures, unlike homogeneous platforms is often NP-Hard. Additionally, compilers, application libraries and operating systems services can vary across architectures. Another major challenge is to make the heterogeneous architecture transparent to application programming and to simplify the process of getting applications to run on the architecture.

One such major challenge for heterogeneous architectures and the focus of this thesis is the communication (both data and coherence messages) across heterogeneous components and the shared system resources of on-chip interconnection network and memory/cache hierarchy. Throughout this work, we use CPU/GPU HSA to demonstrate communication challenges in heterogeneous architectures and our contributions to overcome these challenges. However, most of these challenges are equally applicable to any heterogeneous system. In the context of this thesis, communication in heterogeneous architectures is considered to be broadly comprised of data and coherence control communication.

1.3 Communication Challenges in HSA

Communication and interconnection topologies are one of the major performance and power bottlenecks for architectures that integrate many cores, whether homogeneous or heterogeneous. Heterogeneous architectures pose additional communication challenges due to the differences in computing modalities of each architecture. The major of these being the highly diverse ways in which each of these architectures request for data and

communicate with other cores/memory. These different and potentially conflicting traffics on the shared interconnection backbone lead to interference and suboptimal performance among the integrated architectures. Other challenges arise from presence of shared data between different computing architectures (coherence), inefficient memory sharing, application kernel partitioning and communication policies, etc. Our contributions in this work address the major concern of architectural diversity for communication in heterogeneous systems and improve data and coherence communication performance. In the subsequent sections, we provide a detailed discussion of communication and coherence challenges in heterogeneous architectures.

1.3.1 Data communication challenges in heterogeneous architectures

Networks-on-Chip (NoCs), following packetized communication overcome the communication challenges of many core processors. However, most NoCs designed for homogeneous systems lack the capability to handle diverse traffic requirements of heterogeneous architectures. The diversity in traffic requirements arise from architectural or operational heterogeneity viz.,

- GPUs in HSA operate on thousands of parallel threads and require high throughput, while CPUs are typically latency sensitive and require low latency to achieve superior performance. Homogeneous NoCs, designed to cater to either of these needs, fall short in providing necessary performance for HSA, severely penalizing CPU and/or GPU application performance.
- Presence of shared resource like last level cache, memory controllers, etc. further exacerbates the interconnection challenges, especially for memory intensive applications running on either architecture.
- Integrated subsystems in HSA are likely to be implemented at different technology nodes and operate with different voltage and frequency specifications. Tying them to a single interconnection topology requires additional interfaces for data transfer between different subsystems leading to additional area, power and energy overheads.
- Regular network topologies require efficient node mapping techniques to map heterogeneous cores to network routers, which can have a significant impact on the overall performance of the system. Any difference in integrated architectures, their sizes and operating specifications requires node mapping to be redone completely.
- Finally, off-chip interconnection in multi-chip or chiplet architectures adds high latency and energy overheads for remote and memory communication.

With processing architectures shifting from computation to communication centric systems, these challenges considerably increase the design complexity of interconnection topologies in HSA and impact major performance metrics like inter-core and memory access latency, bandwidth and energy.

1.3.2 Cache coherence communication challenges

Many core systems, homogeneous or heterogeneous, enable parallel computing predominantly through shared memory systems and require coherence protocols to ensure correctness of data shared across multiple cores. Hardware based protocols like directory and snoopy have gained wide spread implementation due to their robustness and superior performance. However, as system size increases, traditional ways of implementing these protocols do not scale efficiently in terms of both performance and overheads. Snoopy protocol faces challenges due to high bandwidth requirements and the need for ordered interconnection support like bus, which typically scale poorly to many cores. On the other hand, directory protocol adds unscalable area and energy overheads with flat directories for accurate sharer tracking beyond few tens of cores. The network latency, access latency and waiting time for directory storage also increases as system size increases. Architectural and operational heterogeneity in heterogeneous architectures further adds to the protocol complexity of cache coherence. In addition to protocol level limitations, coherence control messages further aggravate communication bottlenecks of many core systems described before. Finally, multi-chip and chiplet (multi-die) architectures add another dimension to coherence scalability challenges as addition of central lookup storage or snooping across costly off-chip interconnection is practically infeasible. An ideal scalable cache coherence thus adds low overheads, low network traffic and minimal power consumption and receives hardware support from NoC for optimal transfer of coherence messages.

While many solutions have been proposed to overcome both communication and coherence challenges of heterogeneous architectures, several challenges still persist owing to the ever increasing demands and complexity of these systems. In this work, we propose novel interconnection network and coherence protocols centred around on-chip wireless links, that address the major challenges of communication diversity on shared network and coherence communication in heterogeneous architectures and provides a heterogeneity agnostic design interconnection and coherence protocol that leverages it. The first of these is the HyWin interconnection, built around customized networks and wireless links. HyWin decouples different architectures from one another to minimize interference among them and improve performance across all architectures. Besides improving communication performance, HyWin reduces overheads in presence of operational heterogeneity and reduces design complexity when extending to different heterogeneous architectures. It is robust and easily scales from heterogeneous single chip processors to multi-die and multi-chip systems, irrespective of the integrated ar-

chitectures. Utilizing the HyWin architecture, we further optimize performance through SpyDir and WiSH protocols to provide scalable and high performance coherence. SpyDir is a hierarchical hybrid coherence, that combines local directory with global snoopy protocol to minimize coherence storage and energy overheads, while providing necessary performance at any system size. WiSH further reduces coherence overheads by incorporating application sharing characteristics through logical share aware segmentation of caches.

1.4 Research Contributions

The contributions of this thesis are primarily divided into three categories; an intra/inter chip wireless based interconnection for heterogeneous systems, a hybrid snoopy-directory protocol and wireless channel modelling and link design for use in proposed topology. They are summarized as below

1.4.1 Design of efficient interconnection for heterogeneous architectures

To overcome the challenges of diverse communication paradigms in HSA, we propose intra/inter chip HyWin interconnection, an energy efficient, scalable and unified solution for interconnecting heterogeneous multi-chip components in CPU/GPU heterogeneous system architectures. HyWin topology consists of customized and sandboxed sub-networks interconnected using a system wide hybrid wireless network. Each sub-network in HyWin by itself, is an intra-NoC topology that sandboxes one homogeneous subsystem of HSA from the network/application constraints of other subsystems. The homogeneous subsystems are clusters of similar processing cores and/or memory elements and can range anywhere from few cores in a chip to an entire chip in the system. The sandboxed topologies decouple different architectural cores from global interconnection network and one another and allows more flexibility in designing efficient topology for each subsystem without impacting other sub-networks. Interconnecting the different sub-networks of HyWin requires high performing and efficient interfaces to seamlessly bridge sub-networks with different topologies and protocols across single or multiple chips. HyWin achieves this through a system wide hybrid wireless network comprised of millimeter wave wireless links. It uses a combination of omnidirectional and directional wireless links to provide interference free communication within and among all the chips, while meeting the performance requirements of each architecture. The directional wireless network provides sustained high bandwidth communication for GPU architectures, while non-directional network provides low latency, low energy communication across the heterogeneous system. Furthermore, wireless links, operating at radio frequencies, establish uniform interconnection between subsystems operating at any voltage/frequency or communication protocols. Combining this hybrid wireless network with customized sub-networks, HyWin provides high performance and energy efficient

interconnection for single/multi-chip HSA.

1.4.2 Design of scalable and efficient coherence protocol

To achieve the targets of scalable coherence protocol, we propose WiSH, a wireless enabled, share-aware hybrid coherence that exploits the hierarchical nature of HyWin topology. WiSH coherence is (i) a hybrid Snoopy over Directory (SpyDir) protocol (ii) implemented using HyWin interconnection and (iii) complemented by incorporation of application sharing characteristics into protocol design. The primary aspect of WiSH is the novel SpyDir protocol, which is the hierarchical combination of local directory with global snoopy protocol. Traditionally, hybrid coherence schemes have been implemented with snoopy protocol operating locally within a small core cluster and directory protocol for global coherence. SpyDir, in contrast, implements directory locally within each subsystem and global snoopy for inter-subsystem coherence and is the first such implementation to the best of our knowledge. The ordered interconnection support for snoopy protocol is provided by the wireless network of HyWin, which provides scalable, broadcast network for global snooping. SpyDir leverages the advantages of both directory and snoopy protocols, while complementing the limitations of either with the other to provide low overhead and high performance protocol in heterogeneous architectures at any system size.

The coherence performance and overheads are further optimized in WiSH through incorporation of application memory sharing characteristics into SpyDir protocol design. Memory accesses at each core in a parallel application exhibit recurring sharing patterns that allow for optimizations to improve coherence performance or overheads. WiSH incorporates these access patterns to aid SpyDir protocol by identifying blocks as shared within or between subsystems and privately cached data at runtime. This is achieved by augmenting SpyDir protocol with a logical, dynamic share aware cache segmentation, that holds coherence information only for shared accesses across different cores based on sharing status of memory blocks. The protocol state machine is accordingly modified to implement the share aware cache segmentation and coherence maintenance. Through optimal design of wireless network (link design, number and placement of transceivers), subsystem size and sub-networks, WiSH with hybrid SpyDir protocol provides efficient and scalable coherence for several hundreds of cores.

1.4.3 Wireless channel modelling and link design

The millimeter wave wireless links and network forms the backbone for the proposed HyWin interconnection and consequently WiSH coherence. Hence, efficient design of wireless links and communication infrastructure is of paramount importance. To facilitate this, we perform a comprehensive analysis of wireless channel char-

acteristics in intra-chip environments and design a robust, fading resilient and high bandwidth wireless link for use in proposed topology. To achieve a detailed understanding of on-chip wireless propagation characteristics with near field and multipath propagation effects, we develop a three-dimension model of the complex chip geometries with metal interference structures. We employ transient domain-based solvers to simulate propagation between antennas in these intra chip environments and estimate channel path loss, delay spread and propagation delay. We also present a novel way of implementing on-chip antennas to improve their return loss and bandwidth and study the impact of antenna placement and directionality on channel performance. Utilizing the derived channel characteristics, we propose an Orthogonal Frequency Division Multiplexing (OFDM) based wireless transceiver and link to achieve channel resilient high bandwidth communication in HyWin topology. The OFDM transceiver specifications are set to overcome adverse channel impact and a mostly digital design achieves low overhead implementation with high spectral efficiency.

1.5 Organization of Thesis

The remainder of this thesis is organized as follows:

1. In Chapter 2, we present a detailed discussion on architectural heterogeneity based data communication diversity and associated challenges in single chip CPU/GPU architectures and the intra Chip HyWin interconnection for overcoming the said challenges to provide efficient communication performance.
2. Chapter 3 details the communication challenges in multi-chip heterogeneous system architectures and extends the concept of HyWin interconnection to multi-chip systems through a hybrid combination of omnidirectional and directional wireless networks for inter-chip communication.
3. Chapter 4 discusses the challenges of coherence communication in many core systems and the WiSH cache coherence, which leverages the wireless based hybrid interconnection of HyWin to implement SpyDir protocol and share aware cache segmentation to improve coherence performance and minimize overheads.
4. Wireless channel modelling and link design in Chapter 5 discusses the need for comprehensive analysis of intra chip channel for accurately estimating the performance of wireless links, which form the backbone for proposed architectures. Utilizing this understanding we propose OFDM based resilient wireless infrastructure with high spectral efficiency and low overheads.
5. Chapter 6 finally concludes the contributions of this thesis and discusses future directions of our work and their extension to other heterogeneous architectures.

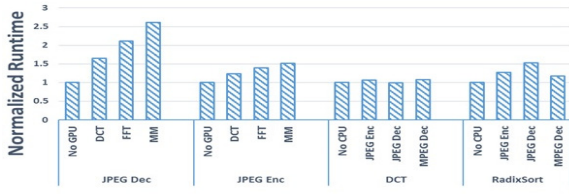
Chapter 2

Efficient Interconnection in Single Chip HSA

Heterogeneous System Architectures (HSA), that integrate CPUs and Graphic Processing Units (GPUs) in a single system have emerged as the most widely implemented class of heterogeneous architectures for improving performance of general purpose parallel applications. State-of-the-art processors like AMD APUs[5, 6, 7] and Intel Core architectures (Sandybridge and later)[8, 9, 10, 11], etc. implement HSA in single chip systems. Integration of GPUs with CPUs is still an evolving concept and faces many challenges for cohesive design, unlike homogeneous systems. In this chapter, we focus on the communication challenges and resource contention in network resources in single chip HSA and discuss HyWin interconnection to overcome these challenges.

2.1 Communication Challenges in Single Chip HSA

Network-on-Chip (NoC) is the communication backbone of the system interconnecting all its components and hence is the most important shared medium in multi-core processors. However, most existing NoC topologies, designed for homogeneous multi-core systems lack the capability to handle diverse traffic patterns and volume exhibited by HSA. GPUs, with many parallel threads require high throughput, while superscalar CPU cores are latency sensitive to achieve superior single thread performance. Traditional NoC resources designed for parallel applications favor GPUs, penalizing CPU application performance, especially with applications running simultaneously on both CPU and GPU. Fig. 2.1a shows degradation in CPU and GPU application runtime in presence of another application running simultaneously on the other processing subsystem. The runtime results are obtained on a system with 4 multi-threaded CPU and 32 GPU cores (specifications discussed in section 2.4). Though application runtime in both cases degrades as compared to the respective application running standalone, CPU applications get more affected than GPU applications. For JPEG encode and decode



(a) Performance degradation of CPU (GPU) applications with GPU (CPU) applications simultaneously executing



(b) Additional Network Energy Consumption due to different voltage/frequency of CPU and GPU

Figure 2.1: Impact on application and network performance due to heterogeneous elements on chip

applications in presence of a GPU application, runtime goes as high as $2.5\times$, when compared with only the CPU application running. A similar analysis in [13] shows that IPC of CPU applications reduces by as high as 84% in presence of GPU applications. Presence of shared memory resources like Last Level Cache (LLC) and memory controllers further exacerbates the problem, especially for memory intensive applications. In addition, with many heterogeneous systems being built from reusable building blocks, there is a fundamental shift from computational to communication centric design methodology. Consequently, on-chip networks play a major role in achieving high performance in HSA. Homogeneous network topologies like simple mesh are inefficient in handling the complex and conflicting communication requirements of HSA and cannot provide optimal performance in terms of bandwidth and energy consumption.

Recently, heterogeneous wired NoCs have been proposed to address application and traffic heterogeneity in multi-core designs[14, 15, 16, 17, 18]. These topologies, in general, use combination of differing network resources like buffered + bufferless routers, packet + circuit switching, etc. and employ different techniques to improve performance. Though such designs improve over homogeneous topologies, they do not consider heterogeneity in different architectural cores and so are not efficient for HSA. Heterogeneous NoC topologies in [19, 20, 21] take into account architectural heterogeneity while designing the network, but still tie them to single regular network topology, limiting the performance. Besides, it is more likely in HSA that, integrated subsystems operate at different voltage/frequency and are from multiple vendors with different technology nodes. So, they require additional network interfaces for data transfer from one subsystem to another. This increases overheads, while making system design complex to balance performance and overheads for efficient interfaces. For example, Fig. 2.1b shows additional packet energy consumed due to network interfaces like level shifters[22] and frequency synchronizers[23] for different CPU/GPU applications. It can be observed that the contribution of these elements to total network energy is significant and increases for systems of larger sizes. In addition, regular network topologies require efficient mapping algorithms to map different heterogeneous cores to network routers, which can have a significant impact on communication delay and energy. This is especially a challenging problem in HSA with shared resources, impacting performance metrics like memory

access latency and bandwidth. To overcome such challenges, we propose HyWin, a hybrid NoC topology with on-chip Wireless Interfaces (WIs) and sandboxed networks for HSA. The proposed topology consists of customized sub-networks interconnected using a system wide wireless network. Each sub-network by itself, is an intra-NoC topology that sandboxes one subsystem (cluster of similar processing cores and/or memory elements) of HSA from the network/application constraints of other subsystems. All such sub-networks are then interconnected with a second level network using millimetre (mm-) wave wireless links that provide low latency and energy efficient interface among them.

The sandboxing topologies of HyWin decouple processing elements of different architectures from system wide interconnection network with single topology. This allows for more flexibility to design heterogeneous networks that are better suited to the communication needs of each subsystem. For example, mesh topology with buffered routers provides high throughput required for parallel execution of GPUs, but is not required for mostly serial execution of CPU cores. A bus based architecture provides low latency communication required for few out-of-order CPU cores, while also minimizing network resources. Interconnecting these different sub-networks requires high performing and efficient interfaces between different topologies and network protocols. The system wide wireless network of HyWin achieves this by providing uniform interconnection between subsystems, irrespective of their topology. WIs convert data from digital domain to radio signals for transmission between two subsystems. WI converts data operating at subsystem voltage/frequency to carrier frequency, that is uniform across the systems and hence avoids the need for any conversion. Further, data transmission through wireless link is serial and data from one subsystem protocol like flit size, packet format, etc. can be converted to another without the need for additional control information. Combined with the performance and energy efficiency of wireless links, on-chip WIs using radio propagation can transmit data seamlessly and interconnect all sub-networks irrespective of their operating voltage, frequency and protocols. Along with efficient communication backbone, wireless channel in HyWin methodology provides a bus like interconnection mechanism among subsystems. This eases application programming with heterogeneous elements, while allowing scalable networks at subsystem level. Combined with customized sub-networks, HyWin provides high performance and energy efficient interconnection for HSA. The main contributions of this work are:

- HyWin topology with sandboxed heterogeneous sub-networks that provide flexible and efficient communication infrastructure for HSA.
- Low latency and low energy interface between different sub-networks based on mm-wave WIs, that avoids the need for frequency/voltage/protocol conversions.
- Faster and high bandwidth access between cores and LLC/main memory in HSA, that scales easily with

number of subsystems and size of subsystems.

The remainder of this chapter is organized as follows: Section 2.2 discusses previous works on different heterogeneous NoC topologies; in Section 2.3, we discuss the HyWin topology along with proposed design for CPU/GPU heterogeneous systems; Section 2.4 describes the experimental setup and results. Section 2.5 provides a summary of proposed work and conclusions.

2.2 Related Work

There have been a wide range of regular homogeneous NoC topologies like Mesh, Torus, k-ary n-cube networks used for multicore systems. These topologies fall short in performance for complex traffic scenarios of HSA. Heterogeneous networks, being explored for such systems combine different router architectures to build upon regular topologies and meet communication requirements of HSA. Authors in [14] make case for HeteroNoC using two router types with different buffer resources to leverage non-uniform network demand in CMPs. They show that providing high buffer resources for routers along the diagonal of a mesh network achieves maximum benefits in terms of performance and power. A heterogeneous NoC design with two different networks optimized for bandwidth and latency steers applications to appropriate network based on its characteristics[24]. Evaluation of topology for different benchmarks shows 6% speedup in application execution, on average for 128 node system. Irregular NoC (INoC) design methodology is proposed in [15] to exploit prior knowledge of communication characteristics and develop customized optimal network topology and routing algorithm for a given design. They use graph algorithms to map different application tasks to NoC routers and design irregular networks best suited for the application. Experimental results show that INoC has at most half the latency as compared to mesh, torus and spidergon topologies at 0.5 packet injection ratio.

An adaptive virtual channel partitioning mechanism has been proposed in [19] to improve bandwidth sharing between CPU and GPU cores for HSA. The proposed technique effectively shares virtual channel resources between GPU and CPU packets to improve system throughput by 15% on average. iConn[20] has proposed an efficient buffer sizing and sharing algorithm for area constrained NoC in heterogeneous architectures. The proposed method improves IPC in CPUs by 23% and GPUs by 9.4%, on average over topology that uses uniform buffer resources. Authors in [21] have explored various placement policies for CPU & GPU cores in ring interconnection for HSA to find optimal network configuration. Using empirical studies, they have shown that enhanced ring interconnection provides maximum performance improvement of 22% in presence of multiple CPU/GPU applications. d-HetPNOC[25], a photonic heterogeneous NoC with dynamic bandwidth allocation

has been proposed for HSA and improves bandwidth by 10%. Jieming et al. in [26] propose a NoC design that exploits slack in GPU packets to route data through non-minimal paths and improve energy efficiency without impacting the performance. They propose flexible pipeline architecture along with DVFS to achieve this and show that energy is decreased by upto 20% with less than 5% performance impact. Same authors in [16] propose Time Division Multiplexing (TDM) based hybrid switching technique to switch between circuit switched and packet switched routing to improve performance in CPU/GPU heterogeneous systems. This method reduces upto 17.1% of network energy with 1.6% CPU performance penalty and 2.6% GPU performance improvement. A unidirectional flow control mechanism for heterogeneous NoCs with both buffered and bufferless routers in fused CPU-GPU architectures is proposed in [27]. They show that IPC performance can be improved by upto 14.1% with only 5.3% energy penalty. Similarly, other heterogeneous NoCs have been proposed to improve network performance for heterogeneous systems and non-uniform traffic scenarios[28, 29].

All these topologies use different techniques to extend the capabilities of regular NoC topologies to meet the diverse communication requirements of HSA. Though efficient, they require optimal routing algorithms, node mapping techniques and network interfaces limiting the performance benefits achieved. To meet the complex communication needs of heterogeneous systems in an efficient manner, we propose HyWin topology in this work. We build customized sub-networks for each subsystem in HSA, which sandboxes them from system wide network; thereby allowing us to meet different traffic requirements optimally in terms of energy and performance. The proposed design then interconnects different subsystems using mm-wave on-chip wireless network. Millimeter wave WIs have been shown to be energy and performance efficient for long distance chip communication[30, 31, 32, 33] and provides efficient interface between subsystems that avoids additional resources. The combination of customized sub-networks for diverse traffic and wireless inter-subsystem network for global performance achieves low latency and energy efficient communication for heterogeneous systems.

2.3 HyWin NoC Topology

We first discuss the design of HyWin topology, in general for any heterogeneous system before presenting its implementation in CPU/GPU HSA and impact of HyWin topology on application and network performance.

2.3.1 Hybrid WNoC with Sandboxed Networks

Heterogeneous systems, in general are composed of few CPU cores, shared memory elements and other different processing elements. These other processing elements can be GPU cores, FPGAs or programmable

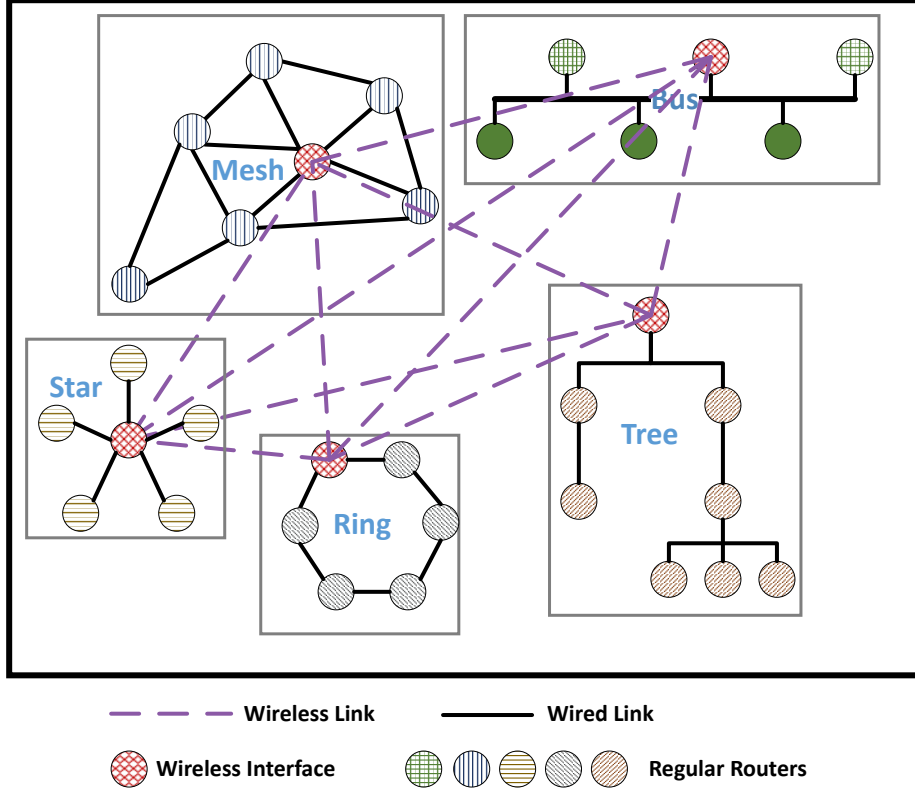


Figure 2.2: Illustrative HyWin Topology with Heterogeneous Sub-networks Interconnected by mm-wave WIs

accelerators to handle domain specific workloads. The concept of HyWin topology to interconnect these different computational resources in heterogeneous systems is shown in Fig. 2.2. In the proposed topology, entire heterogeneous system is divided into multiple subsystems, where each subsystem is a cluster of similar computational components and their private memory elements designed to perform certain class of applications. A customized *sub-network* is implemented for each subsystem that interconnects all its processing elements and memory nodes as identified by different sections in Fig. 2.2. Each of these subsystems have specific traffic patterns and communication requirements like low latency, high throughput, broadcasting, etc. Besides these, some subsystems may require other features such as simultaneous resource use and Quality-of-Service (QoS). The topology of each sub-network is optimized to meet respective subsystem requirements and is completely self contained for all traffic within the subsystem. The router architecture, routing and arbitration are all designed to provide optimal performance and energy efficiency for communication of that subsystem. By implementing sandboxed sub-networks for computational resources of similar type, HyWin topology decouples different heterogeneous system components, allowing for more flexible and efficient network design. This prevents a single network architecture that needs to balance different requirements and penalize one subsystem for another. And most importantly, HyWin significantly improves interconnection scalability and ease of design, where any new computational subsystem can be added to the network without impacting the implementation

of others (like plug and play modules). It also avoids the hassles of optimal node mapping techniques required in single system wide network.

A system wide mm-wave wireless network interconnects different heterogeneous sub-networks of HyWin topology. In each sub-network, at least one gateway node is assigned and provided with WI for communicating with other subsystems. Each WI consists of serializer/deserializer interfaces, transceiver circuit and transmitter/receiver antennas to interface data between digital and radio domains. The inter-subsystem wireless network acts as backbone to provide low latency infrastructure for communication among sandboxed sub-networks. When the source and destination of a packet are within one subsystem, the packet is routed using its sub-network routing protocols. If they are in different subsystems, the packet is routed to the nearest gateway to be transmitted to the subsystem with destination node using wireless medium. The gateway node in each sub-network is chosen to provide easy access to all other routers in the sub-network. The wireless network, along with providing efficient interconnection between subsystems, also avoids additional control or hardware for data syncing, voltage conversion and protocol conversion between subsystems. All inter subsystem data in HyWin is converted to radio signals from respective source subsystem specifications and transmitted over shared wireless medium. At the receiving subsystem, the data is converted back to digital domain operating at corresponding specifications. Hence, the system wide wireless network provides uniform interconnection across all sub-networks and is discussed in detail in section 2.3.4.

2.3.2 CPU/GPU Heterogeneous System Architecture

The CPU/GPU heterogeneous system architecture implemented in our work is shown in Fig. 2.3. The HSA, commonly implemented in modern day Intel/AMD processors, consists of traditional out-of-order superscalar CPU cores and graphic execution units with their private caches, shared cache and memory controllers. The superscalar x86 cores vary from two to ten in number and have very high single thread performance, efficient for serial tasks of the system. Each CPU core supports two threads (similar to hyper-threading[34] in Intel cores) and is associated with *L1* cache (instruction and data) and a single larger *L2* cache is shared among all the CPU cores.

The GPU cluster, on the other hand, is made up of large number of execution units that are more suited for highly parallel applications like gaming, image/video processing, etc. Unlike CPU cores, GPU execution units are comprised of many in-order Single Instruction Multiple Data (SIMD) processors and many threads execute same instruction simultaneously with different data (Throughout this work, we use GPU execution unit and GPU core interchangeably). In our architecture, the entire GPU cluster is divided into multiple *slices*, with

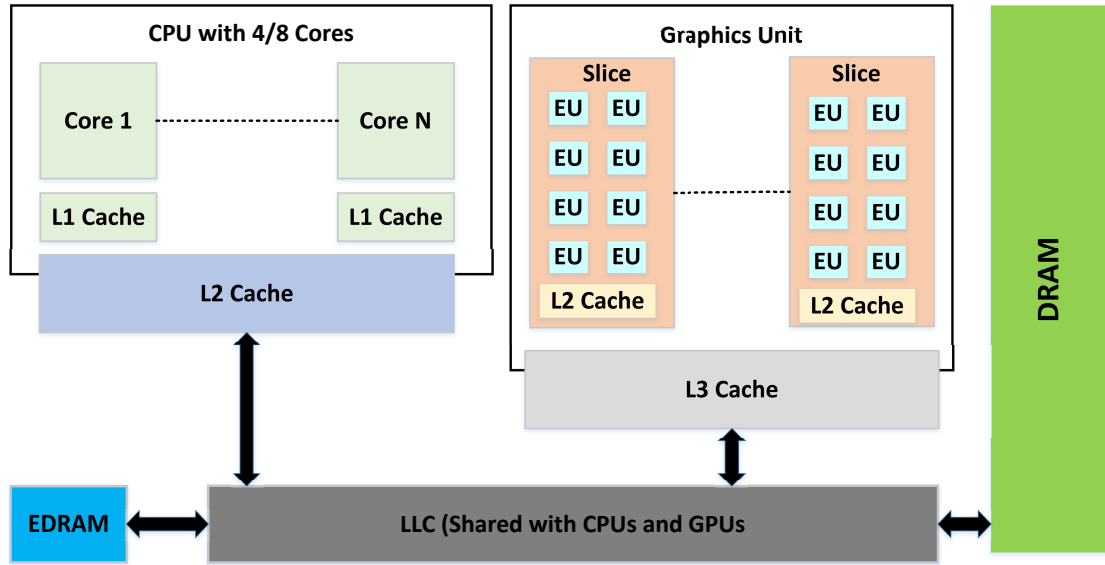


Figure 2.3: CPU/GPU HSA with Shared Memory Implemented in Our Work

each *slice* made up of fewer number of execution units as shown in Fig. 2.3. Each execution unit has its own private *L1* cache, with both instruction and data caches. Each *slice* has a *L2* cache, which is shared by all execution units in the *slice*. Finally, a large *L3* cache is shared by the entire GPU cluster.

A *Last Level Cache (LLC)* acts as the lowest cache before accessing main memory and is shared by both CPU and GPU subsystems. An off-chip *DRAM* module acts as the main memory that holds all the required data for CPU/GPU operations. A large off-chip cache, often implemented as embedded DRAM (*eDRAM*) is sometimes included on the package and can be used as another level in memory hierarchy before accessing the main memory. The exact sizes of each cache, main memory, number of CPU/GPU cores, etc. are discussed in the experimental evaluation section.

2.3.3 HyWin Topology for CPU/GPU HSA

HyWin topology implemented for CPU/GPU HSA is shown in Fig. 2.4. As described before, the high traffic volume injected by GPU cores interfere with CPU applications in HSA, resulting in performance degradation. To achieve these conflicting network requirements of CPUs and GPUs while avoiding performance impact, entire system is divided into four subsystems; CPU subsystem, GPU subsystem, shared Cache subsystem and main memory subsystem. The CPU subsystem is comprised of all CPU cores, their associated *L1* caches and shared *L2* cache. The primary requirement of CPU cores is low latency access to lower level caches and main memory. This can be perfectly achieved using *bus based* sub-network for the few CPU cores by connecting all

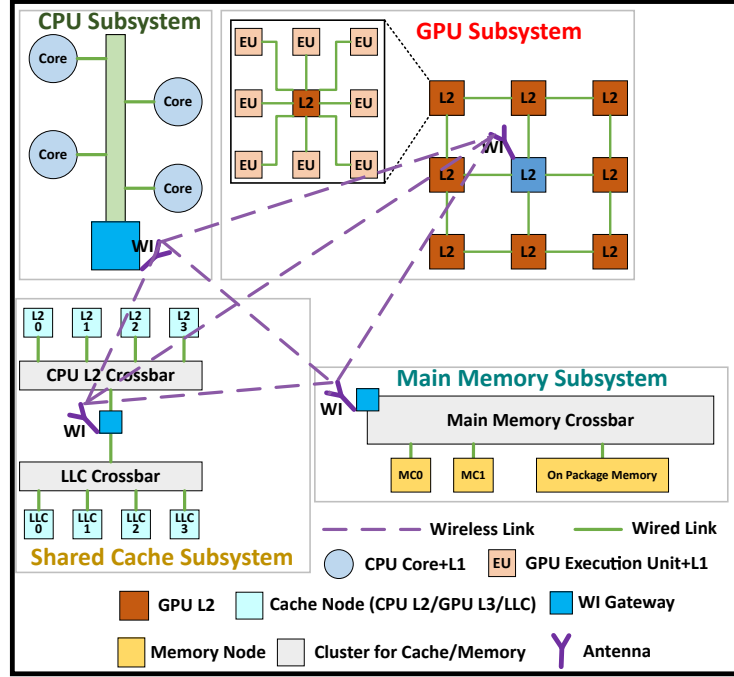


Figure 2.4: HyWin Topology for the Implemented CPU/GPU HSA

L1s directly to shared bus medium as shown in Fig. 2.4. The CPU applications follow serial execution and share minimal data between the cores. CPU cores use large caches and a variety of complex speculative mechanisms to obtain desired program data from shared cache and main memory. Hence, the network infrastructure for CPU subsystem is not burdened by frequent packet injections but, needs to provide faster memory access when required. The bus architecture meets all these requirements when number of cores is small and provides low latency access to shared CPU L2 cache. Any data, if needs to be communicated among the cores, is easily accessible to all cores through shared bus. An arbiter is associated with bus architecture that controls access to shared medium for data requests from/to each core. A WI is attached to the bus that handles all data communication with LLC, GPUs and main memory (discussed in detail in later part of the section).

The GPU subsystem in HSA, in general is used for gaming and general purpose parallel tasks. The number of execution units in GPU subsystem is different in each architecture and can, in general, vary anywhere between 16 and 80 execution units. GPU subsystem in the proposed topology consists of all GPU execution units, their respective L1 caches and shared L2 caches. GPUs use massive multi-threading to cover high memory latency and executes other threads in place of threads stalled for memory instructions. Thus, GPUs generate high volume of traffic and inter-core data, thereby requiring very high throughput networks. Homogeneous NoC topologies used for CMPs are designed to deliver high network throughput and are ideal for interconnecting GPU cores and their caches. The GPU cores in most system sizes evaluated in this work are interconnected using *star-mesh* topology as shown in Fig. 2.4. The *star-mesh* topology provides sufficient bi-sectional band-

width for GPU systems with moderate to high number of cores. The GPU cores and shared L2 cache of a slice are connected in a star topology with L2 acting as center of the star network. All shared L2s from each slice are then connected in *mesh* topology. The routers used for different cache nodes are input buffered to provide required high throughput. Any two communicating cores within a slice require at most two hops. Communication with a core in another slice requires the number of hops equal to sum of Manhattan distance between respective L2 nodes and two more hops from L2 to source/destination core. A WI is implemented at the center of mesh topology as shown in Fig. 2.4 to act as gateway for communication with remaining subsystems in the system. The gateway is equidistant and easily accessible from all cores in the GPU subsystem.

For systems with smaller number of execution units in GPU subsystem, simple mesh network is sufficient to achieve the required bandwidth. As number of execution units increases, a simple mesh or star-mesh fall short of providing the necessary performance. For larger GPU subsystem sizes, more efficient network topologies like WNoC[30, 31, 32, 33] is implemented for GPU sub-network. In such case, WNoC sub-network of GPU transmits data over a frequency channel that is different from that of system wide wireless network. The details of HyWin with intra-WNoC sub-network for GPU subsystem are discussed in scalability subsection of 2.4. The sandboxing of HyWin allows redesign of GPU sub-network without affecting the topology or requirements of other subsystems. This considerably reduces the complexity in implementing HyWin topology for different systems and makes it easily scalable.

The remaining two subsystems of the HSA are shared cache and memory systems, which hold the necessary program data and whose architecture varies significantly from system to system. Along with shared caches for each subsystem, a last level cache is shared between both CPU and GPU cores. The shared cache subsystem is comprised of this LLC and shared L3 cache of GPU subsystem. The presence of shared memory resources between CPU/GPU can increase the impact of interference between applications. Each shared cache in subsystem is connected to the sub-network through crossbar topology. Hence, there are two crossbars; GPU L3 and LLC crossbars are shown in Fig. 2.4. A shared cache (e.g., GPU shared L3) is provided with multiple nodes that service different address ranges and all such nodes connected to the respective crossbar router. The crossbar provides high bandwidth and single hop access to all nodes in the memory. The crossbar routers from different caches are then connected to the WI gateway to service memory requests from CPU/GPU and access main memory. Any data to be shared between GPU L3 cache and LLC is serviced through WI gateway router. Finally, the main memory subsystem of HSA is comprised of the memory controllers, that service memory requests to read/write data from main memory (DRAM). The memory controllers are connected using a crossbar topology and WI gateway is provided with the main memory crossbar, that acts as interface to entire system. Emerging processor designs like [35] implement an off-chip embedded DRAM on the package for higher per-

formance. In such cases, the *eDRAM* module is connected to crossbar, similar to one of the shared cache levels. The wireless link provides low latency access to this off-chip memory, further improving its performance.

The customized sub-networks sandbox different subsystems and provides optimized communication in each subsystem while not interfering with others. These different sub-networks must be interfaced with system wide network to provide communication between subsystems. Especially, the second level network must provide low latency and high bandwidth access to memory and shared caches, which is achieved using mm-wave wireless links as shown in Fig. 2.4. The designated gateway routers in each subsystem are provided with WI, comprised of serializer/deserializer (SERDES), transceiver (Low Noise Amplifier (LNA), modulator/demodulator, Power Amplifier (PA)), transmitting and receiving antennas[30]. These WI components act as interface for packet data between wired network and wireless medium. The transceiver uses *On-Off keying (OOK)*[30] to modulate packet data stream for wireless transmission, which is chosen for its low area overheads. The shared wireless medium is implemented using single channel, that allows one packet transmission at any time. The single channel communication is implemented using *zigzag* antenna and token management controller[30] is used to provide access to shared wireless medium. Single channel wireless communication is chosen to maintain consistency with baseline WNoC topology (used for comparison and explained in section 2.4). The performance can be further improved by use of multi-channel wireless links[30]. By using combination of low latency wireless network and sandboxed sub-networks, HyWin meets heterogeneous traffic requirements of HSA.

2.3.4 WIs as Interconnection Backbone

The wireless network of HyWin topology, besides providing low latency and low energy system wide interconnection, also offers seamless and efficient communication interface between different subsystems.

- **Data Syncing:** Different subsystems in HSA often operate at different clock frequencies, requiring some form of data synchronization while transferring data from one subsystem to another. This is especially difficult, when techniques like DVFS are used. Wireless network avoids additional synchronization hardware by use of SERDES circuit in each WI, which act as interface between parallel data of wired network and serial data stream of wireless network. Since, all WIs operate at same antenna frequency, data written to or read from WI buffers at any clock frequency is converted to serial data stream with same bit period. Hence, data from subsystems of different clock frequencies is inherently synchronized without additional hardware, as long as SERDES circuit is designed to operate at highest possible system clock frequency. The size of the buffers is same as that of the ones used at the input ports of the router. They are also useful in maintaining protocol consistency, when different subsystems use different protocols like packet

format, control signals, etc.

- **Voltage Conversion:** Another issue, similar to different clock frequencies, is operating voltage of different subsystems. HyWin methodology mitigates this by appropriate configuration of PA and LNA of WI transceiver in each sub-network. These circuits are configured to allow digital data at respective subsystem voltage to be transmitted as reliable radio signal over wireless medium. This, otherwise, requires many voltage level shifters at all router ports interfacing between two different subsystems. By avoiding the additional hardware of data synchronizing and level shifting at all interfacing routers of existing topologies, WI reduces significant area overhead while providing the required network performance.
- **Miss penalty:** One of the important performance metric in processor architectures is miss penalty in accessing lower level caches or off-chip memory. Large caches and main memory have high access latency, limiting the performance of computation when desired data is not present in core L1 caches. This is further increased by network latency in many core and heterogeneous architectures. Wireless links, with their low latency communication over long distances across the chip, reduce miss penalty significantly as compared to wired links. The delay savings are significant for accessing LLC and off-chip memories, thereby minimizing processor idle time waiting for program data. The low latency and high bandwidth of wireless links in accessing off-chip memory can further be exploited to reduce the number of cache levels in memory hierarchy.
- **NoC Node Mapping:** Heterogeneous systems, in general, require efficient node mapping techniques that map different processing or memory elements to NoC routers. This mapping has direct impact on communication latency and energy between any two cores/memory. The proposed topology avoids need for such techniques as sub-network connecting components of one subsystem forms a homogeneous intra network by itself. This avoids need for router mapping in a sub-network, generally required when using single topology across entire system. The task mapping algorithms of software interface is sufficient to assign application tasks to desired processing elements to achieve optimal performance.

2.3.5 Routing Algorithm

The routing algorithm to transfer data between any two communicating nodes in heterogeneous system using proposed topology is shown in Algorithm 1. In HyWin topology, each sub-network uses its own routing strategy to route data within the sandboxed topology. When both source and destination nodes are within the same subsystem, data is routed using the respective routing algorithm of the sub-network. When the source and destination nodes are from different subsystems, data from the source is first routed to WI gateway using the

Algorithm 1 Routing Algorithm

Initialize: Assign wireless token to CPU subsystem
At any router except gateway:
while Request for output **do**
 Decode destination address
 if Address within subsystem **then**
 Use intra routing strategy
 (eg., XY for GPU, bus arbitration for CPU)
 else
 Find the shortest path to WI
 (eg., Manhattan distance in GPU, single hop in CPU)
 end if
end while
At WI:
while Data to be transmitted **do**
 Serialize and modulate the flits
 if token available **then**
 Transmit data using wireless medium
 After tail flit is transmitted, pass token to next subsystem in order
 else
 Wait for token
 end if
end while
while Data received **do**
 Decode destination address
 Use intra routing strategy to reach destination
end while

routing algorithm of the source subsystem. At the WI gateway, the data is modulated and transmitted to the WI gateway of destination subsystem. The data from receiving WI is routed to final destination using the routing algorithm of that subsystem. The wireless medium connecting different subsystems is a shared medium and a control mechanism is required to provide access of wireless channel to each WI. A token management controller with round robin arbitration is used to grant access to shared wireless medium and is adopted from [30]. The token is passed in the following order: CPU subsystem, GPU subsystem, shared cache and main memory. At power on, the token is first assigned to CPU subsystem to load initial system applications. When data is available at WI gateway, it waits for token to transmit this data. While WI is waiting to get access to the token, the flits to be transmitted are stored in buffers at the interface. The size of the buffers in WI is same as that of the buffers at wired ports. When it receives the token, the WI starts transmitting the data and token remains with the transmitting WI till a single packet is transmitted. Once the tail flit is transmitted, the WI relinquishes control of token and it is passed to the next WI in loop.

In the HyWin topology implemented for CPU/GPU HSA, GPU subsystem uses XY algorithm for routing

Table 2.1: Details of CPU/GPU Heterogeneous System Architecture

Component	Configuration
CPU	x86-64, 4 Out-of-Order (OOO) cores with 2 threads each, $2.5GHz$, 4-way, 1-cycle, $32KB$ L1I cache, $32KB$ L1D cache, 8-way, 4-cycle, $256KB$ L2 cache, LRU policy, $128B$ line
GPU	AMD Southern Island, 32 cores with 8 slices (4 cores in each slice), $1.5GHz$, 4-way, 1-cycle, $32KB$ L1I cache, $32KB$ L1D cache, 8-way, 4-cycle, $256KB$ L2 cache, LRU policy, $128B$ line, $3MB$ L3 cache
LLC	8-way, 20 cycle latency, $16MB$ LLC
Memory	$2048MB$ memory with 2 controllers
Coherence	MOESI directory protocol

data within the GPU cluster. If GPU needs to communicate with either of the other subsystems, data is routed to the WI using XY routing and then transmitted to respective subsystem as described. The CPU subsystem is implemented using bus based architecture and when a core needs to communicate, it just needs to place the data on shared bus. The data can be immediately accessed by appropriate destination cores or WI gateway. The routing in CPU subsystem is controlled by bus arbiter, which controls bus access to different cores. Each core gets access to the bus using round robin arbitration. For data communication with other subsystems, data from/to any core is transferred to WI gateway attached to the shared bus. In both shared cache and main memory subsystems, whenever data read/write request is received, the request is transferred to the node servicing appropriate address range. The node acknowledges the read request along with requested data and is transmitted to CPU/GPU using WIs. An acknowledgment is transmitted back in case of write request. If requested data is not available in shared cache, the request is forwarded to main memory and the data from main memory can be transmitted back directly to the core requesting the data. Simultaneous memory requests can be serviced by the use of buffered routers at cache/memory nodes.

2.4 Experimental Results

In this section, we discuss the experimental setup, architecture of CPU/GPU HSA, details of HyWin topology and its performance evaluation and overheads.

2.4.1 Experimental Setup

To evaluate performance of HyWin topology for CPU/GPU HSA, we use Multi2Sim[36] heterogeneous system simulator, that accurately models different CPU and GPU architectures. Multi2sim is modular, cycle accurate and allows configuration of CPU/GPU cores, their specifications and memory hierarchy. It also allows flexi-

Table 2.2: Details of HyWin Topology for CPU/GPU HSA

Topology	Bus-based for CPU, Star-Mesh for GPU, Crossbar for LLC/Memory
Routers	5-port, 3-stage (RC, SA, ST), 8-flit depth per port
Link	64-bit width, 1 cycle latency
Bus	4-flit width, round robin arbitration
Wireless Link	60GHz carrier, 16 Gbps bandwidth, 1 cycle latency
Routing	XY routing for GPU

bility of specifying interconnection network and routing algorithm for implemented system making it ideal to evaluate the proposed topology. The general configuration of CPU and GPU in the implemented HSA is shown in Table 2.1. Each thread in CPU core has its own L1 cache and single L2 cache is shared between both the threads. Each GPU slice has four execution units with their private L1 caches. All execution units within the slice share a single L2 cache. The L3 cache is shared among all GPU slices. We also evaluate scalability of proposed topology with different system sizes. The evaluated system sizes are chosen to match existing designs like Intel’s core processors and future trends of HSA. Application level traffic and statistics are obtained by simulating different CPU and GPU benchmarks or combination of both. Multi2sim tool provides pre-compiled binaries of SPEC2006[37], OpenCL[38] and MediaBench[39] benchmarks.

The configuration of HyWin designed for the implemented heterogeneous architecture is shown in Table 2.2. As described before, the topology uses shared bus for CPU, star-mesh for GPU and crossbar for memory elements. The private L1 cache of each CPU is directly connected to shared bus medium and it uses round robin arbitration to service multiple requests from different cores. Star-mesh topology of GPU uses 5-port router with three pipeline stages of Route Computation (RC), Switch Allocation (SA) and Switch Traversal (ST). All wired links between the routers are 64 bit wide and have single cycle latency. The routers are synthesized using TSMC 32nm technology in Synopsys Design Compiler to compute the area and power overheads. The single channel wireless links use 60GHz carrier frequency and have bandwidth of 16Gbps. The WIs are granted access to shared wireless medium by using the token based approach[30]. The WI components are synthesized in Design Compiler using 65nm technology.

To compare the performance of HyWin, we use baseline mesh and WNoC topologies for the implemented architecture. The baseline mesh topology for the implemented architecture is shown in Fig. 2.5. The CPU tile consists of CPU core, associated L1 caches and L2 cache. The GPU tile represents single slice with four GPU cores and their shared L2 cache. Each tile is provided with a router. The placement of different cores/memory of HSA in mesh network is done such that CPUs/GPUs have distinct paths to LLC and memory controllers. This is done to minimize congestion and contention for shared memory and is shown to provide better performance in [19]. WNoC topology augments baseline mesh with on-chip wireless links operating at

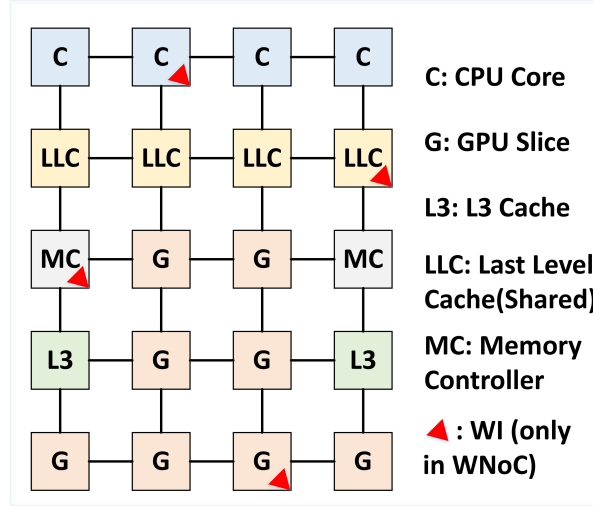


Figure 2.5: Baseline Mesh Topology for Implemented HSA

Table 2.3: Benchmark Characteristics for CPU and GPU Applications

CPU Benchmarks			GPU Benchmarks		
Benchmark	Suite	Size/Characteristics	Benchmark	Suite	Size/Characteristics
BZip	SPEC2006	2 MB files with three levels of compression	Binary Search	OpenCL	Array length of 500000
Cactus	SPEC2006	100 iterations with uniform grid size of 0.1	DCT	OpenCL	256×256 matrix
DealII	SPEC2006	8 refinement cycles	EigenValue	OpenCL	Diagonal matrix of 512 length
GCC	SPEC2006	Train test set	Fast Walsh	OpenCL	Array length of 50000
GSM De-code	MediaBench	260 bit compressed speech	FFT	OpenCL	NA
GSM En-code	MediaBench	160 13-bit speech frames	Gaussian	OpenCL	5 images with sizes in range of 512×512 to 1536×1536
JPEG De-code	MediaBench	229×147 JPEG image	Matrix Multiply	OpenCL	Two matrices of 256×256 size
JPEG En-code	MediaBench	229×147 RAW image	Matrix Transpose	OpenCL	512×256 matrix
MPEG Decode	MediaBench	4 frame MPEG video	Radix Sort	OpenCL	Array length of 100000
MPEG En-code	MediaBench	4 frame raw video format	Sobel	OpenCL	512×512 image

mm-wave frequencies. The characteristics of wireless links used in WNoC topology are same as those used for inter-subsystem communication in HyWin. A simulated annealing based optimization algorithm[30], that minimizes average hop count of the network, is used to determine the number and placement of WIs in WNoC

topology. Using this algorithm, implemented WNoC topology consists of four WIs placed at routers 2, 8, 9 and 19 (router 1 is top left router in baseline mesh) as shown in Fig. 2.5. Both mesh and WNoC topologies use input buffered 5-port routers with three pipeline stages. The width of links between the routers is equal to 64 bits. The routing in baseline mesh is implemented using XY algorithm and routers with WIs in WNoC use *NorthLast* routing strategy. *NorthLast* routing strategy is adopted from SouthLast routing of [40], which shown to be deadlock free in presence of wireless links.

The application scenarios are chosen to be CPU only, GPU only and concurrent execution of both CPU and GPU applications. This allows us to evaluate impact on network performance when both CPU and GPU subsystems are active and contend for shared resources. The experimental evaluation includes (i) 10 CPU applications, (ii) 10 GPU kernels and (iii) exhaustive combinations of both CPU and GPU applications. The characteristics and sizes of the simulated benchmarks are shown in Table 2.3. CPU benchmarks are taken from SPEC2006 and MediaBench suites and contain both integer and floating point applications. GPU benchmarks, from OpenCL, include matrix, search, sort and signal transform applications. These benchmarks represent most of the commonly occurring real life application scenarios. Each CPU/GPU benchmark is simulated 10 times and average metrics over all runs are considered for evaluation.

2.4.2 Performance Evaluation

To evaluate the performance of HyWin topology and understand the improvements achieved over regular mesh network, we compare application runtime, average memory latency and peak bandwidth of both topologies.

2.4.2.1 Application Runtime

The application runtime improvements using proposed topology over baseline mesh and WNoC are quantified by speedup in execution time of CPU thread(s) and GPU kernel (if present). An application is comprised of both computation and memory instructions and application speedup is dependent not only on number of memory accesses, but also on distribution of memory accesses throughout the application execution. The application speedup with HyWin topology over baseline mesh and WNoC topologies for CPU, GPU and CPU+GPU applications are shown in Fig. 2.6 (For sake of figure clarity, a subset of CPU+GPU combinations is shown in Fig. 2.6c, but similar trend is observed across all combinations). The x86 and GPU results for each topology correspond to improvement in CPU runtime and GPU runtime respectively. In the presence of CPU only applications, no contention for resources exists. The speedup is primarily provided through faster access to caches/main memory and is determined by number of memory requests in the application. GSM, GCC and

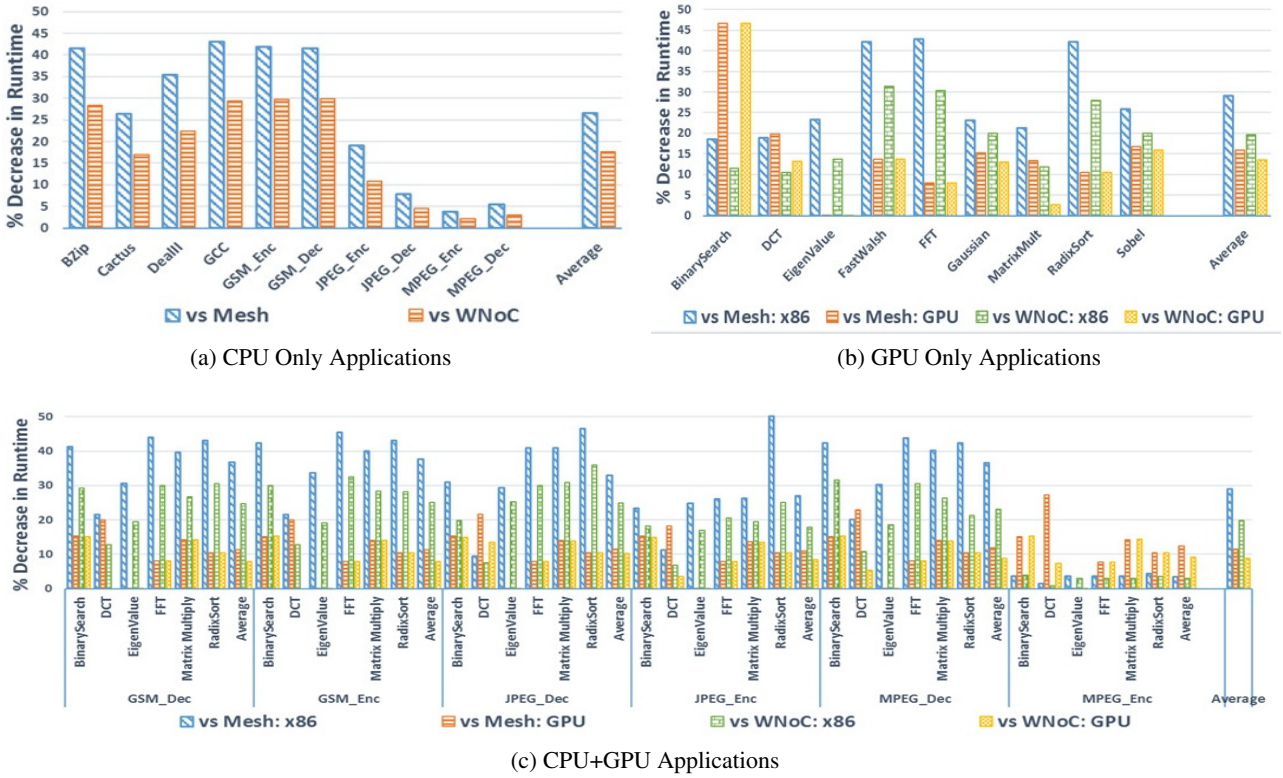


Figure 2.6: Improvement in CPU and GPU runtime using HyWin topology over baseline mesh and WNoC for different application scenarios

BZip applications, as shown in Fig. 2.6a, provide runtime speedup of over 40% over baseline mesh, while speedup in other applications is moderate to little. On average, HyWin improves runtime by 26.6% and 17.7% over baseline mesh and WNoC topologies respectively. When both CPU and GPU are active simultaneously and contention for resources increases, the advantages of sandboxed sub-networks and long range wireless links become more apparent. This is observed from GPU only (Fig. 2.6b) and CPU+GPU applications (Fig. 2.6c). The GPU applications are comprised of one master CPU thread and GPU kernel. The x86 runtime improvements shown in Fig. 2.6b correspond to the execution of CPU master thread, while GPU runtime shows improvements in kernel execution. With CPU+GPU applications, HyWin topology provides application speedup ranging from 10% to 50% for CPU threads and 2% to 35% for GPU kernels with average speedup of 29% and 11.6% respectively over baseline mesh topology. In comparison with WNoC topology, HyWin provides average runtime improvements of 19.8% and 8.8%, on average in CPU and GPU threads respectively. HyWin provides twice the amount of improvement in application runtime over mesh as compared to WNoC, primarily achieved by avoiding network resources sharing between both subsystems. In addition, the experimental evaluation setup utilizes same compiler optimized for existing system architectures which do not account for long range wireless links, limiting the benefits achieved with HyWin topology. The reduction in application runtime can be further improved by including low latency and high bandwidth memory access across characteristics provided by WIs

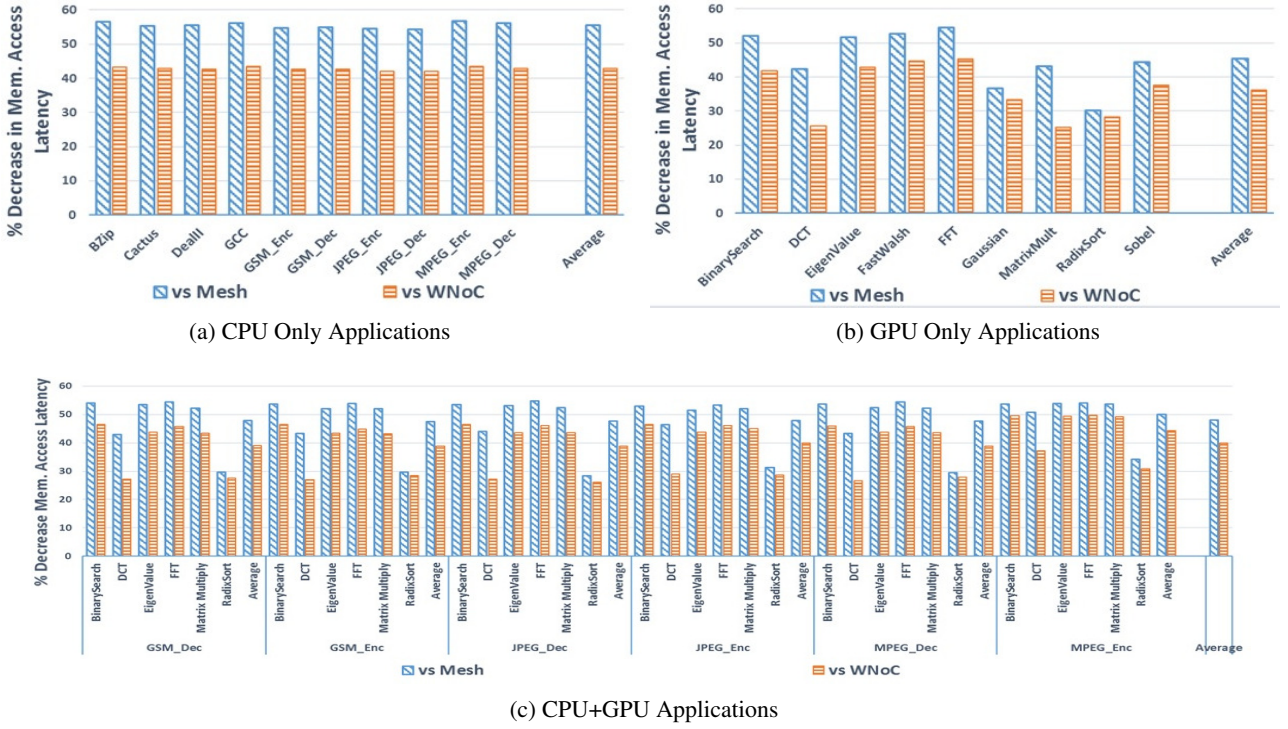


Figure 2.7: Memory access latency in cycles using Mesh, HyWin and WNoC topologies over baseline mesh for different application scenarios

into compiler optimization.

2.4.2.2 Memory Access Latency

Another important metric in processor performance is the memory or LLC access latency, that acts as major bottleneck for obtaining instruction or program data. This further degrades performance in presence of shared memory resources in HSA. Fig. 2.7 shows decrease in average memory access latency provided by HyWin over baseline mesh and WNoC topologies for different application scenarios. Across all applications, HyWin topology reduces memory latency by $\approx 50\%$ and $\approx 40\%$ in comparison with baseline mesh and WNoC topologies respectively. Furthermore, mapping of CPU/GPU cores to network routers in mesh/WNoC topologies determines the number of hops required by each core to access memory controllers and has a significant impact on memory latency. The results for mesh and WNoC in this work are obtained by achieving optimal mapping of resources that minimizes the average communication cost of the network and has non-overlapping paths to memory for CPU and GPU cores. Placement of cores in network with non-overlapping memory paths has been shown to provide the best performance in HSA[19]. Though WNoC topology also has wireless links that provide low latency across long distances on chip, the sub-network architecture of hybrid WNoC provides faster access to WIs further reducing the latency. For CPU only applications, saving in latency is further higher, lim-

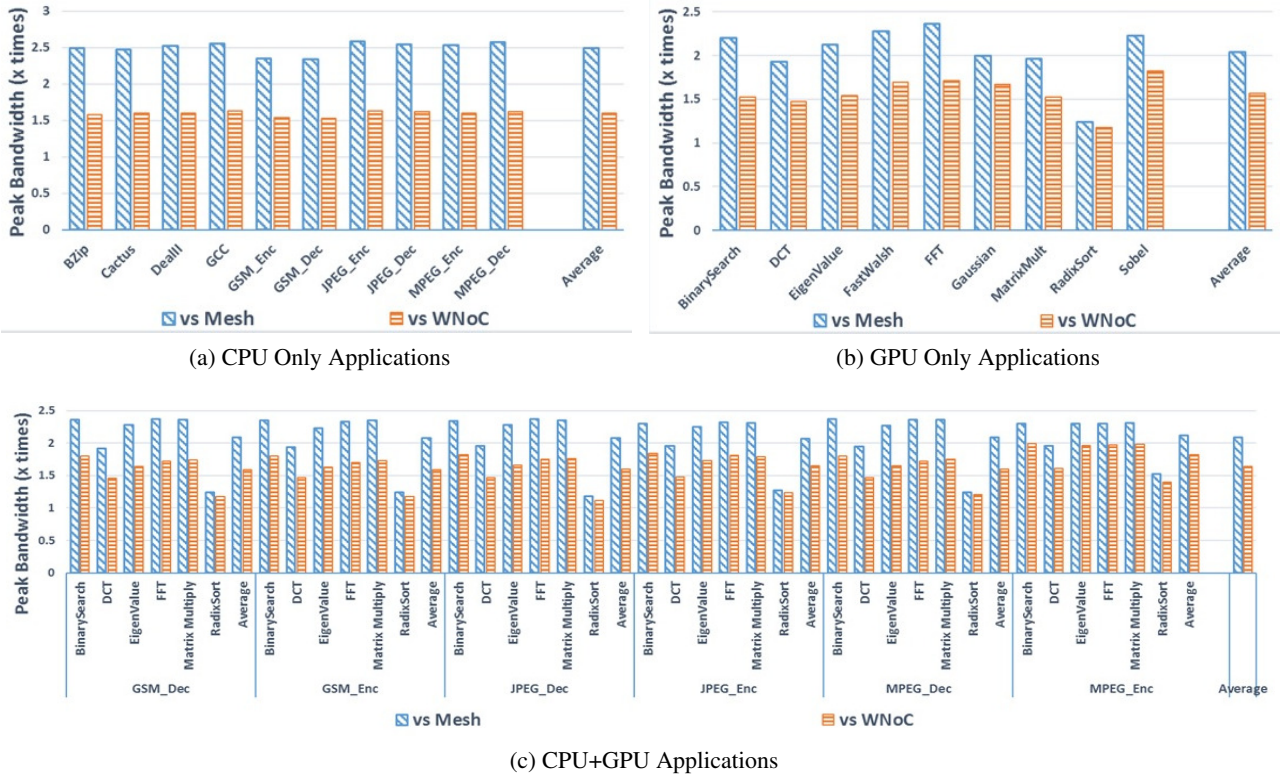


Figure 2.8: Improvement in peak network bandwidth using HyWin topology over baseline mesh and WNoC for different application scenarios

ited only by memory requests passing through multiple cache hierarchy levels. Even with both CPU and GPU accessing memory, no degradation in latency performance is observed with HyWin, whereas average latency increases by 12% in case of baseline mesh topology. Several works have tried to address the shared memory resource challenges in HSA using variety methods[13, 41]. The memory performance can be further improved by including such techniques with proposed HyWin topology.

2.4.2.3 Bandwidth

We evaluate the peak bandwidth to understand the maximum throughput provided by the proposed topology and compare the values with that of mesh and WNoC topologies. Fig. 2.8 shows the improvement in peak bandwidth provided by HyWin topology over baseline mesh and WNoC topologies for all application traffic. HyWin topology provides $2.1\times$ and $1.6\times$ more bandwidth as that of mesh and WNoC topologies. Though WNoC architecture provides similarly good improvements in bandwidth for CPU only applications, the advantages of HyWin become significant with mix of CPU+GPU applications and proposed topology has 60% higher bandwidth on average.

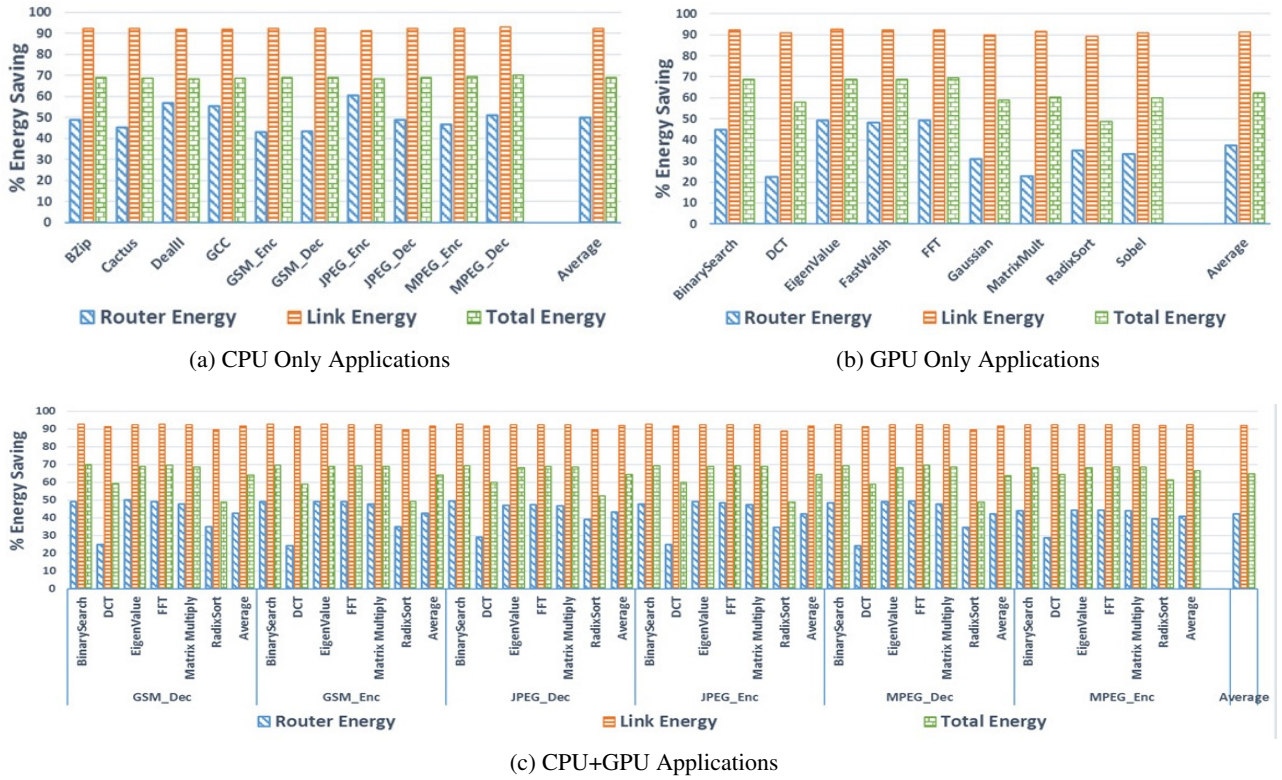


Figure 2.9: Savings in average packet energy using HyWin topology over baseline mesh for different application scenarios

2.4.3 Energy Savings

Fig. 2.9 shows the savings in average packet energy in the network using HyWin topology over baseline mesh topology. As observed from the figure, proposed topology saves 42.1% and 91.8% of router and link energy respectively in comparison with mesh topology. The reduction in router energy is primarily achieved by the optimal design of network resources in each sub-network. The baseline mesh topology uses buffered routers at every processing/memory node and has significant power consumption, both dynamic and static, especially in the CPU subsystem. HyWin topology, on the other hand, uses bus with only an arbiter and WI in CPU subsystem and crossbar topologies in lower cache hierarchies, minimizing router energy significantly. The energy consumption within GPU subsystem (mesh sub-network) is similar in both topologies because of the similarities in network configuration. For servicing the memory requests, HyWin uses wireless links to access LLC and main memory, which cover long distances across the chip, thereby reducing link energy significantly as compared to mesh that uses multiple hops for accessing memory controllers. Combined with optimal router resources, end-to-end energy consumed to service any read/write request is considerably smaller. In total, the proposed topology saves 64.5% of the packet energy over baseline mesh across all application scenarios. Use of low power techniques like dynamic scaling for network routers[42] and power gating for WIs[43] further

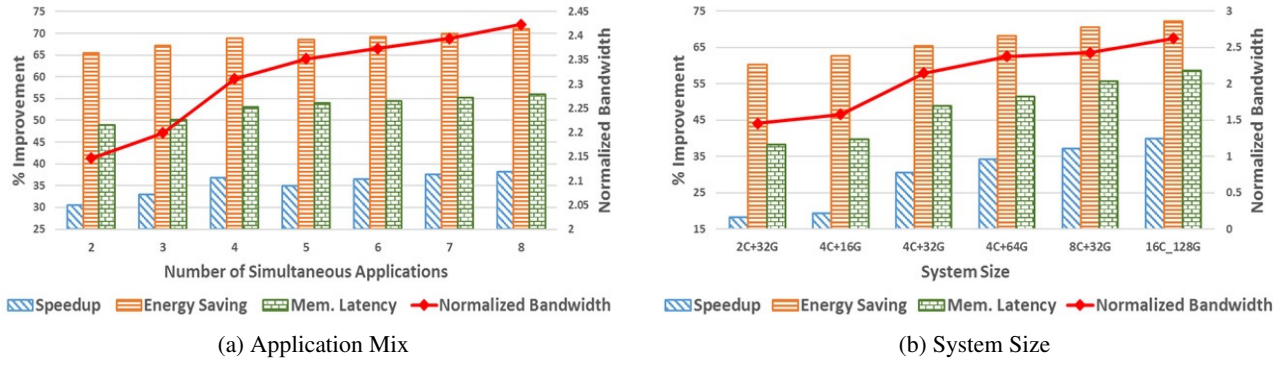


Figure 2.10: Scalability Analysis of HyWin Topology

improves the energy efficiency of the proposed topology.

2.4.4 Scalability

The sandboxing feature of customized sub-networks in HyWin, along with long range wireless links make the proposed topology easily scalable. We evaluate the scalability of HyWin in two categories; number of different applications running on an architecture and system size of the architecture. A number of applications running simultaneously on a given system results in complex memory access patterns and inter core communication, stressing the system to its limits. The scalability of HyWin in terms of applications is evaluated by stressing the system with increasing number of CPU and GPU applications. The number of simultaneous application running on the system is varied from two to eight (Multi2sim does not allow context switching after an application is assign to CPU thread and hence maximum number of applications is limited to number of CPU threads). In each case, all exhaustive combinations are simulated and average performance metrics are obtained. For example, scenarios with 5 applications include 5C, 5G, 1C+4G, 4C+1G, 2C+3G and 3C+2G, where 'C' and 'G' are number of CPU and GPU applications. The average speedup, energy savings, decrease in memory access latency and normalized bandwidth for different number of applications mixes is shown in Fig. 2.10a. The improvements calculated are over baseline mesh topology. As can be observed from the figure, performance improvements provided by HyWin increase with increasing number of applications. As number of simultaneous applications increase, only 1% – 5% degradation in performance is observed using HyWin topology as opposed to 10% – 18% with baseline mesh topology.

In second scenario, we observe scalability of proposed topology by increasing system size, in terms of both CPU and GPU cores. We vary the number of CPU cores from 2 to 16 and GPU cores from 16 to 128. With increasing number of cores in CPU or GPU subsystem, topologies like bus and mesh respectively may

Table 2.4: Sub-network Topologies at Different Subsystem Sizes

Subsystem	Size (No. of Cores)	Sub-network Topology
CPU	1-4	Bus
	5-8	Star-Bus
	8-16	Mesh
GPU	16-32	Mesh
	32-64	Star-Mesh
	64-128	WNoC

Table 2.5: Comparison of HyWin Topology with Existing Works

Topology	Performance	Energy
[26]	1-8% decrease in performance of GPU	Upto 20% reduction in network energy
[16]	1.6% decrease and 2.6% increase in CPU & GPU performance	Upto 17.1% decrease in network energy
[27]	14.1% average improvement in IPC	5.3% increase in energy
HyWin	29% speedup; 2× BW; 50% latency	42% & 92% decrease in router and link energy

not provide desired performance. Hence, we implement different sub-network topologies as subsystem size increases as shown in Table 2.4. For GPU subsystems with 64 and 128 cores, GPU sub-network is implemented using WNoC topology for better network performance. The number and placement of WIs in GPU sub-network is done using the simulated annealing based optimization[30]. Using this algorithm, the number of WIs for both 64 and 128 core GPU subsystems is 4. In this case, system wide and GPU subsystem wireless networks are implemented using two non-overlapping channels to avoid interference between both networks. The WIs in intra-WNoC topology of GPU uses dual band PLPA antenna[44], that radiates at 45 GHz and 60 GHz . Data communication within GPU subsystem uses 45 GHz channel, while all communication by GPU with other subsystems uses 60 GHz channel (this frequency is same as that of system wide wireless network frequency used by other sub-networks). The bandwidth provided by the dual band PLPA at both 45 GHz and 60 GHz is almost equal, thereby providing equal performance at both frequencies. Any WI present in intra-WNoC topology of GPU sub-network can be used as gateway for inter-subsystem communication, thereby improving network characteristics for systems with large GPU sizes.

The average improvements over different application scenarios provided by HyWin over baseline mesh at different system sizes are shown in Fig. 2.10b. As can be observed, with increasing system size, HyWin scales better providing better performance as compared to mesh topology. This is because mm-wave wireless links provides low latency communication across the chip, even with increasing size and sub-network topologies are modified to provide optimal performance at given size. In addition, sandboxing feature of HyWin avoids complex node mapping algorithms, which need to be evaluated at each systems size. This considerably reduces effort for scaling the topology to any system and modify one subsystem without impacting others.

2.4.5 Comparison with Existing Works

Table 2.5 shows the comparison of HyWin in terms of performance and energy consumption with existing works. To ensure fair comparison, we include works that report either application or network performance and energy results. The DVFS mechanism with GPU slack exploitation proposed in [26] saves up to 20% of network energy with GPU performance decrease of 1-8%. Their method exploits higher tolerable delays in GPU to use non-minimal paths and reduce network frequency and energy consumption. A hybrid switching technique, in [16], establishes circuit switched networks between frequently communicating nodes, while other nodes use packet switching to achieve up to 17.1% reduction in energy while impacting CPU performance by 1.6%. A heterogeneous topology with buffered and bufferless routers[27] increases IPC by 14% on average with an energy penalty of 5.3%. In comparison, HyWin provides application speedup of 29% on average. It provides $2\times$ the bandwidth of baseline mesh and 65.5% saving in average packet energy. The combined advantages of customized sub-networks with wireless links significantly improves both performance and energy characteristics.

2.4.6 Overheads

The four stage router with 8-flit deep buffers at each input port in the GPU sub-network of proposed topology and baseline mesh topology consume an area of $7.67 \times 10^{-3}mm^2$ including all the components. The router implemented with each cache/memory node requires an area of $2.32 \times 10^{-3}mm^2$. The area of cluster routers connected at each cache level and memory is depended on number of nodes and has an area of $16.6 \times 10^{-3}mm^2$. The transceiver circuit including OOK modulation, LNA and PA requires a total area of $0.3mm^2$ per WI. HyWin topology for the implemented CPU/GPU HSA, including all components requires 17.39% less area as compared to baseline mesh topology. HyWin saves area in CPU subsystem, where four stage regular routers are replaced by bus architecture and bus arbiter. The WI including transceiver and serializer/deserializer buffers consumes a total power of $32mW$. Each router implemented in GPU sub-network consumes $4.46mW$ and $6.44mW$ of dynamic and static power consumption respectively. The total power consumption in routers at each cache/memory node and cluster is $4mW$ and $20.24mW$ respectively. The transceiver circuit of WI consumes static power of $39.646mW$ including all of its components. HyWin topology, while requiring 17.36% area lesser than baseline mesh topology, provides 50% and 60.5% improvement in latency and energy for CPU/GPU HSA. The wireless interconnection backbone with sandboxed sub-networks can be scaled to any design with different subsystems.

2.5 Conclusions and Future Work

A heterogeneous NoC topology, HyWin using on-chip mm-wave wireless links is proposed in this work to meet the diverse communication requirements of heterogeneous systems architectures. The proposed topology implements heterogeneous sub-networks for each subsystem, interconnected by a system level wireless network. HyWin topology speeds up application by 29% for CPU and 12% for GPU on average in the presence of both CPU and GPU applications. The improvements in application performance are achieved by sandboxing different subsystem to avoid contention and conflict for network resources. The long range wireless links of HyWin reduces the average memory access latency by 50% and provides twice the bandwidth as compared to baseline mesh network. In addition to providing high performance, HyWin reduces the network energy consumption by 64.5%, while requiring lesser chip area resources as compared to that of the baseline mesh network.

HyWin provides an efficient interconnection for handling diverse traffic requirements of heterogeneous architectures and achieving the desired performance. However, there are several ways in which it can be further improved in terms of its efficiency and implementation.

- In the current implementation, the wireless interface in each sub-network acts as sole gateway for communication with other subsystems, which impact network reliability in case of a failure in wireless subsystem. Possible solutions like redundancy in wireless infrastructure and alternative communication routes over wired links, etc. help improve the overall reliability of the interconnection.
- As the number of subsystems in the heterogeneous system increase, the single channel wireless network can become a congestion point for performance. Exploration of multi-channel and high bandwidth wireless links designs enhance the scalability of HyWin architecture for larger number of subsystems. OFDM based wireless links discussed in 5.11 and 5.13 offer potential solutions.

In this chapter, we have discussed the communication challenges prevalent in single chip CPU/GPU architectures and HyWin interconnection concept to overcome the said challenges. While, challenges like interference on shared interconnection due to architectural and operational heterogeneity remain in multichip HSA also, additional limitations arise due to remote chip communication and inefficiency of inter-chip links, etc. This requires innovative modifications to HyWin concept to overcome the additional challenges of multichip HSA. We elaborate on these challenges in the next chapter and extend the concept of HyWin to system architectures with multiple CPU and GPU chips.

Chapter 3

Hybrid Wireless Interconnection for Multi Chip HSA

Multichip computing modules with several chips integrated with main memory banks are found in a wide range of platforms from servers to embedded systems. The integrated chips in such systems include processing chips such as multicore CPUs, GPUs, accelerators or a heterogeneous mix of such chips (e.g. nVIDIA's DGX-1[12]) depending on the desired functionality. Application performance in such systems is as highly dependent on efficient interconnection as it is on the efficiency of the processing architectures. Integration of heterogeneous architectures further adds to the interconnection performance and resource management challenges due to diverse traffic patterns of different architectures and their interactions (as discussed in chapter 2). Additional bottlenecks arise in multichip heterogeneous systems due to inefficient inter-chip links, remote communication, shared memory space, etc. To overcome these challenges, we enhance HyWin interconnection with hybrid wireless network infrastructure for inter-chip communication, where each processing chip acts as a sandboxed homogeneous subsystem with its custom intra-chip network topology. In this work, we primarily focus on CPU-GPU multichip systems, which are fast garnering significant interest because of their unique capabilities in handling different workloads. While we discuss the communication bottlenecks in the light of CPU-GPU setups, the limitations remain same for any heterogeneous multichip system.

3.1 Communication Challenges in Multichip Heterogeneous Systems

Evolution and advancements in multichip computing systems shifts processing architectures from computation to communication centric with increasing contribution of interconnection performance to the overall system

performance. Especially, processor-to-memory communication and remote communication between processing chips have a significant impact on the achievable performance. However, inter-chip interconnection in multichip systems faces several challenges viz.,

1. Exponential increase in memory bandwidth requirements of each individual chip due to the significant ramp up in the number of cores on a single chip is a major concern for performance.
2. Increasing prevalence of memory intensive applications like machine learning and IoT further exacerbates the memory bandwidth challenges.
3. The scaling up of number of chips in the multichip system poses difficulties for efficient management of shared resources like interconnection and memory subsystem.

Heterogeneous multichip architectures only further exacerbate these challenges. Besides processor to memory communication, major bottlenecks for inter-chip communication, even in single CPU-single GPU systems, arise from (i) data communication between host CPU and GPU and (ii) remote communication between the chips (CPU/GPU). The former challenge arises from the way CPU-GPU systems have traditionally been integrated using PCIe[45] switches to connect one or more GPUs to host CPU. While GPUs have access to dedicated VRAM for high bandwidth memory communication, host communication and data transfer from main memory through PCIe switches severely limits the performance and has high energy consumption. The significantly different access patterns of GPUs and CPUs (high throughput for GPU and low latency for CPU, in general) and interference between both processors compounds the limitations imposed by existing interconnection infrastructure. Unified memory[46] addressing model helps simplify access mechanisms to an extent, but the underlying hardware interconnection limitations still exist.

The latter problem of remote inter-chip communication is especially a limiting factor in systems with multiple CPUs and multiple GPUs. Though single CPU/GPU setups experience limitations, they are not as profound. In multichip setups, the interconnection network exhibits very high latency, when a requesting core tries to access remote memory locations from another chip. Multiple GPUs sharing same CPU host further limit available bandwidth as all GPUs have to share the PCIe channels and limit scalability for memory intensive workloads[47, 48]. Moreover, each of the heterogeneous system components might operate at different voltages, frequencies and on-chip protocols and will need voltage/frequency or protocol conversion, while communicating with memory modules or other chips in the system. Existing interconnection modalities, based on wired links, for connecting multiple chips fall severely short in providing the desired communication performance requirements of heterogeneous multichip systems. Therefore, a prudent, energy efficient and scalable

interconnection network is required, which can address homogeneous, heterogeneous and memory integration challenges for different multichip environments/systems and provide a unified solution to interconnection issues faced in multichip systems.

Several efforts are under way, from both industry and academia, to address the interconnection challenges for inter-chip communication. In-package memory techniques like 3D integration provides efficient memory communication, however they suffer from high thermal density and also do not solve chip-to-chip communication limitations. Inter chip interconnection solutions like Intel Omni-Path interconnect[49], AMD Infinity fabric[50] in multi-CPU setups and NVIDIA NVLink[12, 51] for multi-GPU setups provide high bandwidth communication in multichip computing systems. However, these solutions address only multi-CPU or multi-GPU setups, whereas we address a similar issue but provide a generic solution that extends to communication between any two chips, either processor or memory.

In this work, we propose to use wireless interconnects to provide an energy efficient, scalable and unified solution for interconnecting a heterogeneous multichip system with multicore processors, GPUs and in-package memory modules. Multi gigabits per second wireless interconnects through on-chip embedded antennas[52, 53] provide the backbone for the proposed chip-to-chip interconnection. It uses a combination of omnidirectional and directional wireless links to provide interference free communication among all the chips, while meeting the performance requirements of each processor. Using directional dual-band on-chip antennas, we propose a wireless interconnection architecture that establishes direct wireless links between GPUs and memory banks, while simultaneously sustaining non-directional single-hop wireless links between processor tiles in different processor chips. The system wide non-directional links provide low latency, low energy communication for CPU-to-memory and CPU-to-CPU/GPU traffic. The dedicated directional wireless links provide sustained, high bandwidth communication desired by GPU architectures. Furthermore, wireless links, operating at radio frequencies, establish uniform interconnection between chips operating at any voltage/frequency or communication protocols. Each processing chip and memory module in the system is equipped with at least one wireless transceiver to enable inter-chip communication. We propose a Medium Access Control (MAC) and routing protocol suitable for hybrid wireless interconnect fabric and evaluate it in comparison to state-of-the-art and emerging interconnect technologies. The major contributions of this work are as follows:

- A hybrid chip-to-chip wireless interconnection using omnidirectional and directional wireless links for energy efficient communication in heterogeneous multichip systems.
- Point-to-point directional wireless links for sustained, high bandwidth communication between GPUs and main memory modules. System wide omnidirectional wireless infrastructure for low latency com-

munication between any two chips.

- MAC and routing protocols for the hybrid wireless interconnection to provide interference free, high performance communication in multichip systems.
- Detailed evaluation of proposed wireless interconnection at different system setups and wireless links implementations.

The remainder of the chapter is organized as follows. Section 3.2 summarizes the state-of-the-art and existing works addressing the inter-chip communication challenges in high performance computing systems. The CPU/GPU multichip system and the proposed inter-chip wireless interconnection architecture are described in section 3.3. Section 3.4 provides details of the wireless infrastructure. The communication protocol for providing interference free, reliable wireless transmission between chip is described in section 3.5. The experimental setup and evaluation of the proposed interconnection are provided in section 3.6. Section 3.7 provides a basic view of different communication behaviours that impact extension of proposed hybrid wireless interconnection to any heterogeneous multichip architecture. Finally, the work is concluded in section 3.8.

3.2 Related Work

The proliferation of multi-chip computing modules and the trend towards heterogeneous computing platforms necessitates efficient and seamless inter-chip interconnection backbone to complement the raw computing power of different architectures. Towards achieving this, there have been several efforts from both industry and academia addressing different aspects of chip-to-chip communication challenges.

3.2.1 State-of-the-Art

Traditional interconnection modalities, according to International Technology Roadmap for Semiconductors (ITRS) (<http://www.itrs2.net/>), have not been scaling at the same rate as on-chip interconnects and gate lengths. This implies a huge gap in density and performance of traditional I/O systems and the ever increasing communication demands of high performance computing systems. The wiring complexity as number of individual chips increases only further exacerbates the problem due to design challenges, crosstalk, and signal integrity issues. State-of-the-art in-package memory techniques like 2.5D (memory on same substrate or interposer)[54] or 3D (vertical stacking of memory over processing die)[55, 56] integration address off-chip memory communication challenges to provide low latency and high bandwidth memory access. However, either or both of

these techniques suffer from high power/thermal density, limited memory size, low reliability and yield due to through-silicon vias, thin wafer sizes, layer misalignment, etc[54, 57, 55]. Additionally, they do not completely eliminate off-chip access and remote chip communication still remains a challenge in multichip systems.

To address the inter-chip communication challenges in multi-socket CPU setups, Intel and AMD have introduced Omni-Path Architecture (OPA)[49, 58, 59] and Infinity Fabric (IF)[50] respectively. Intel OPA, described through a series of layers of network stack, provides low latency, high bandwidth interconnect fabric between CPU and memory components in large scale high performance systems and data centres. AMD IF uses two communication planes, scalable data and scalable control fabrics, that provides coherent interconnection across single chip and multi-socket architectures in any topology[60]. NVLink[12, 51], introduced by nVIDIA, addresses interconnection challenges in multi-GPU setups to provide high bandwidth communication through point-to-point channels between GPUs and CPU (of the order of 5 to 12 times as compared to PCIe). However, these state-of-the-art solutions mostly address only either multi-CPU or multi-GPU setups. Intel's Embedded Multi-Die Interconnect Bridge (EMIB)[61, 62] embeds small silicon bridges and uses microbumps to interconnect multiple dies on a heterogeneous chip package. By avoiding expensive interposer and TSVs, EMIB aims to improve yield as compared to 2.5D technology and interconnect heterogeneous dies (different processing nodes, analog/digital) efficiently. Our work proposes a generic solution that extends to communication between any two chips in single package or multichip systems. Furthermore, limitations of electrical links in these interconnections limit the achievable performance to meet the future demands of high performance computing systems. Towards this end, there are efforts to integrate emerging interconnection technologies for high bandwidth and low power chip-to-chip communication.

3.2.2 Emerging Interconnects

Owing to the limitations of electrical interconnects, various novel interconnect technologies such as photonic [63], inductive or capacitive coupling based[64] and wireless interconnects[65] are being explored to mitigate the performance issues of conventional I/O based multichip systems. Ultra-low power photonic interconnects are envisioned to provide chip-to-chip connectivity in multichip systems[63]. However, pitch of photonic interconnects, laying out of waveguides and precise control of chip temperature profiles, etc. pose challenges to their implementation. Moreover, this technology is challenging to integrate with standard CMOS processes, typically requiring a separate photonic device layer with large footprints on the chip. While [66] has proposed wireless inter-chip interconnection for system with in-package memory, it had not considered a heterogeneous multichip environment and therefore used only an omnidirectional antenna. In [67], physical layer designs for wireless connected multichip modules have been proposed for high performance computing environment.

A comparative study for inter and intra chip communication using emerging technologies has been done in [68]. While these approaches explore implementing emerging interconnects for multichip systems, they do not consider heterogeneous architectures. An interconnection methodology using wireless interconnects to integrate heterogeneous systems involving CPU and GPU cores, but within a single chip, has been proposed in [69]. However, significantly more research needs to be conducted to extend it to overcome the bottlenecks of conventional multichip integration methodologies. Also, multichip systems involving heterogeneous chips are yet to be investigated for an interconnection that can simultaneously handle CPU and GPU traffic efficiently, and provide a unified interface among heterogeneous system components with low integration overhead. In this work, we present a novel wireless interconnection to integrate heterogeneous multichip systems with different processing chips and main memory, and compare its performance with state-of-the-art technologies.

3.3 Wireless Interconnection Architecture for Heterogeneous Systems

In this section, we discuss the CPU/GPU heterogeneous multichip system and the proposed inter-chip wireless network, that provides low latency, low energy and high bandwidth communication between different processing chips and memory modules in a heterogeneous multichip architecture. While we discuss the proposed interconnection in the context of CPU/GPU architectures, it is extensible to any heterogeneous system.

3.3.1 CPU/GPU Heterogeneous System Architecture

The CPU/GPU heterogeneous multichip system consists of multiple many core CPUs, multiple GPUs and main memory banks associated with each chip as shown in Fig. 3.1. The cores in each processor chip is interconnected using a wired intra-chip Network-on-Chip (NoC) topology. The total main memory associated with each processor is comprised of multiple chips. The processors access the memory through integrated memory controllers, distributed across the processing chip. The inter-chip interconnection is provided through conventional chip-to-chip buses and PCIe in baseline and wireless network in the proposed architecture.

3.3.1.1 CPU Intra Chip Topology

Each CPU chip in the system is a many core processor and is modelled Intel Xeon scalable processor platform [70] as shown in Fig. 3.1a. It is comprised of several cores and corresponding cache hierarchy, interconnected using an intra-chip wired mesh topology. Each processing core is based on X86 architecture, capable of handling two threads. The cache hierarchy has three levels, where L1 and L2 caches are private to the core, while

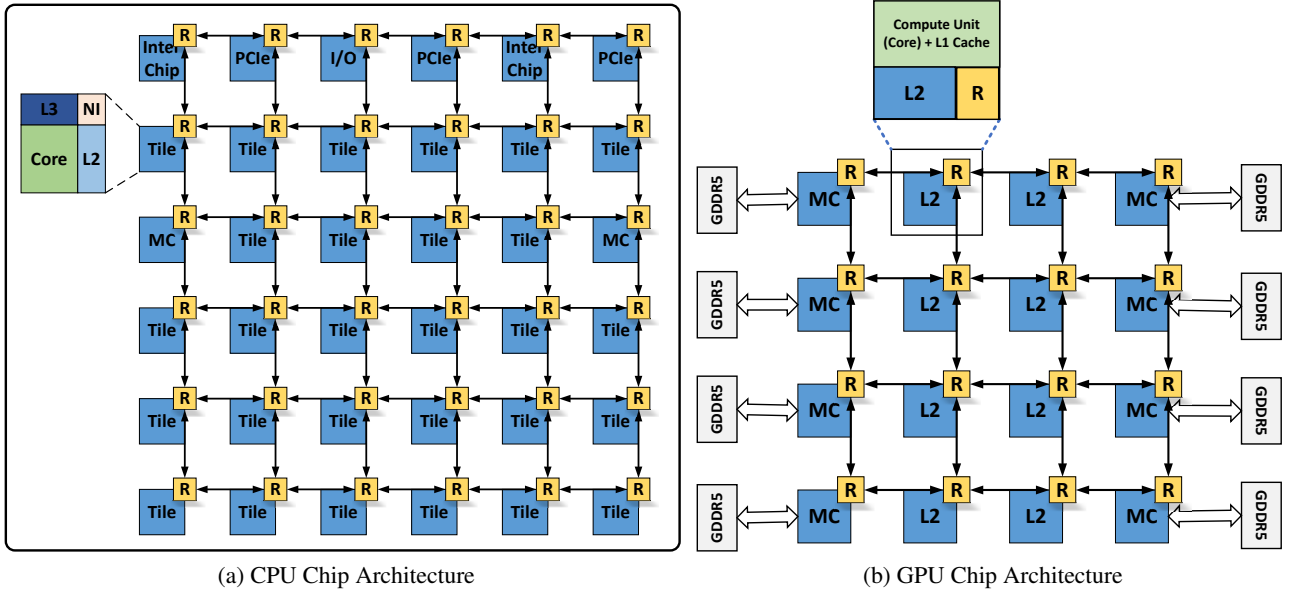


Figure 3.1: CPU/GPU heterogeneous multichip system and intra-chip topologies of different processing chips

L3 cache is shared among all the cores. Each tile in the mesh network is comprised of the CPU core, its private caches (L1 and L2) and a shared L3 cache bank. The tile is fitted with a wired router for interconnecting it with other tiles in the mesh network as shown in Fig. 3.1a. All the routers within the mesh network are connected using 64 bit wired links. Each CPU chip has its associated main memory connected to it through integrated memory controllers in baseline architecture. The intra-chip topology remains same between both baseline and proposed approaches, except for the location of memory controllers and I/O peripherals. They are explained in detail in section 3.3.1.3 and section 3.3.2 respectively.

3.3.1.2 GPU Intra Chip Topology

The GPU chip in the system is modelled after extant architectures from nVIDIA and AMD, where each compute unit is made up of several hundreds of simple scalar processors. Each compute unit has a private L1 cache, shared by all threads running on the compute unit. In the context of this work, we ignore specialized caches like texture and software managed caches. All the compute units in the architecture share a single L2 cache. The L2 cache is divided into multiple banks and are integrated with memory controllers for accessing off-chip graphic memory. In the processing hierarchy, once a kernel is loaded onto the GPU, the corresponding data is copied from main memory to the graphic memory to provide high bandwidth memory access. The compute units and L2 cache banks within each GPU are interconnected using a wired mesh topology as shown in Fig. 3.1a. Each node in the topology is comprised of a compute unit, its private cache and a shared L2 cache bank. The inter-router links are made up of 64 bit wires. The integrate memory controllers are distributed across the

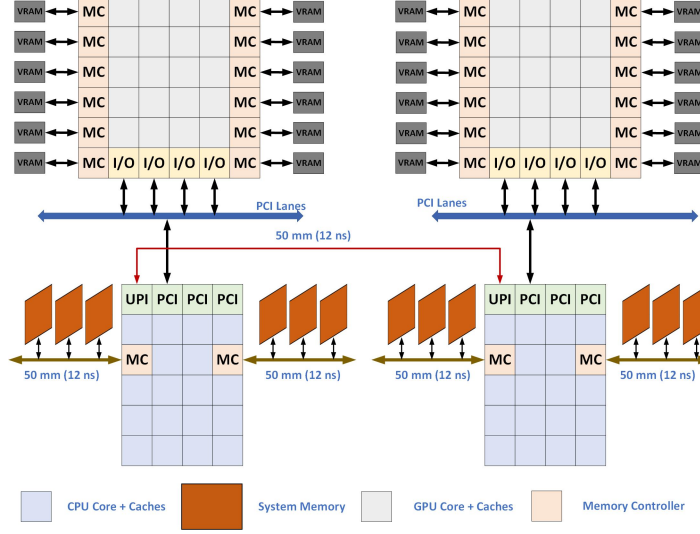


Figure 3.2: Baseline Inter-Chip Interconnection for Heterogeneous Multichip Systems with Buses and PCIe Links

topology to provide optimal bandwidth and energy consumption in both baseline and proposed architectures.

3.3.1.3 Baseline Inter-Chip Interconnection

In this section, we briefly describe the baseline inter-chip interconnection for the CPU/GPU heterogeneous system. The baseline inter-chip interconnection follows the traditional approach of networking high performance multi-socket systems with memory buses, chip-to-chip I/O and links for CPU and PCIe for GPUs as shown in Fig. 3.2. As described earlier, both CPU and GPU chips communicate with associated main memory through integrated memory controllers. The CPU chips, as shown in Fig. 3.1, each use two memory controllers placed along the first and last columns of the mesh following state-of-the-art systems like Intel Xeon platform[70]. Each memory controller is capable of handling upto three channels and communicates with main memory chips through a bus interconnect. The CPU chips communicate with each other through point-to-point wired links modelled after Intel OPA and corresponding controllers are placed along the first row of the mesh topology. The specifications of both memory and inter-CPU links are provided in section 3.6.1. Any CPU chip trying to access remote memory location (memory associated with another CPU chip) has to forward this request through inter-chip links, which adds considerable performance overheads in baseline architectures.

In the baseline architecture, each GPU connects to a CPU chip through PCIe, through the PCIe controller on the CPU chip as shown in Fig. 3.2. Multiple GPUs can be connected to a single CPU depending on the available number of PCIe lanes on the CPU chip. In our implementations, we assume each GPU chip to be using 16 PCIe lanes and the performance of PCIe links is configured to account for different overheads as highlighted

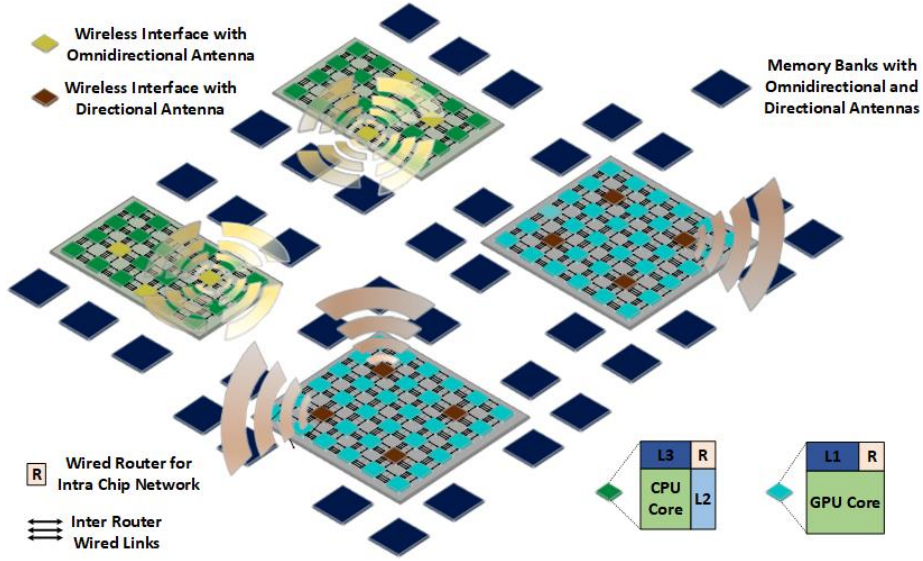


Figure 3.3: Chip-to-Chip Wireless Interconnection with Directional and Omnidirectional Wireless Links for Heterogeneous Multichip Systems

in [71]. Data transfers between main memory and Video RAM (VRAM) is routed through PCIe controllers and main memory controllers on the CPU chip. Remote communication between two GPUs connected to the same CPU uses the PCIe controllers, while communication between those connected to different CPU chips has to be routed through inter-chip links between the CPU. This remote communication between different chips, either CPU or GPU, adds considerable overhead to the overall performance, when an application is executed across multiple chips. Furthermore, bus based interconnection and electrical wire limitations act as congestion points when multiple requests have to be transferred. The proposed inter-chip interconnection overcomes these challenges through high bandwidth, low energy wireless network between any two chips.

3.3.2 Inter-Chip Wireless Network

The inter-chip wireless network shown in Fig. 3.3, in proposed architecture, is designed to meet the diverse memory access patterns of CPUs and GPUs in performance and energy efficient manner, while eliminating the limitations of traditional approaches. CPU cores are traditionally optimized to perform efficiently for serial execution, while GPUs, with thousands of simple scalar threads, work better for highly data parallel applications[72]. Accordingly, CPUs exhibit irregular memory access patterns that require low latency and GPUs have frequent, high bandwidth memory access pattern to support thousands of cores. To meet these diverse requirements, we implement hybrid wireless network, that combines omnidirectional and directional antennas as shown in Fig. 3.3.

The application execution on any single core in multicore CPUs is interspersed by irregular memory access instructions, which stall the execution till the memory request is serviced. Though multi-level caches help improve performance, high off-chip memory access latency can significantly impact application performance. In addition, control messages like coherence requests add to overall execution latency, especially when sharer cores are spread across multiple chips. To provide low latency access to both main memory and remote cores in other CPU/GPU chips, we implement inter-chip wireless network for CPUs using omnidirectional antennas. The omnidirectional wireless network connects each CPU chip with every other processor or memory chip in the system, greatly reducing latency of main memory and remote chip access. This omnidirectional network is implemented on single wireless channel, operating at frequency, f_{omni} and all to and from CPU chips is transmitted over this frequency. In our implementations, we have used $60GHz$ frequency for the omnidirectional network. To enable the omnidirectional wireless network for CPU, all its integrated memory controllers, PCIe switches and I/O peripherals are replaced with Wireless Interfaces (WIs) and zigzag antennas. The WI infrastructure and antennas are described in detail in section 3.4. The minimal impact of minor distance variations on wireless latency allows a degree of freedom on placement of WIs across the chip. Utilizing this, we optimize the placement of WIs in the mesh network to reduce the average hop count to each the nearest WI and further reduce latency as shown in Fig. 3.3. In addition, our evaluations have shown that four WIs across the chip efficiently cater to the communication needs of the system as opposed to a total of six I/O in baseline system to reduce the area overhead of the network.

Unlike CPU architectures, GPUs hide long memory latency by executing an active threadblock, when current threadblock is stalled due to a memory instruction. They run thousands of threads in parallel and generate regular memory accesses, that have steady, high bandwidth requirements. To ensure this type of connectivity, we equip GPUs with directional antennas to form point to point directional, high bandwidth wireless links with their associated memory. Similar to CPUs, the memory controllers on GPUs are replaced with WIs and their placement is optimized to reduce the intra-chip latency for main memory access as shown in Fig. 3.3. The placement of directional wireless links between each GPU and memory is chosen to provide interference free, high bandwidth memory access. As shown in Fig. 3.3, each memory controller on the GPU chip forms directional links with few memory banks. The directional antennas on both memory controller and associated memory banks are aligned to be in the end fire regions of each other to enable seamless connectivity. To ensure interference free and sustained bandwidth over directional wireless network, the radiation regions of the directional antennas are such that (i) the end fire regions of antennas on each link are perfectly aligned with each other and (ii) end fire regions of any two directional links do not have any overlap between them. Hence, the total available bandwidth over the directional wireless network is equal to sum of individual bandwidths of all point-to-point links. It operates at frequency, f_{dir} to avoid interference from omnidirectional network

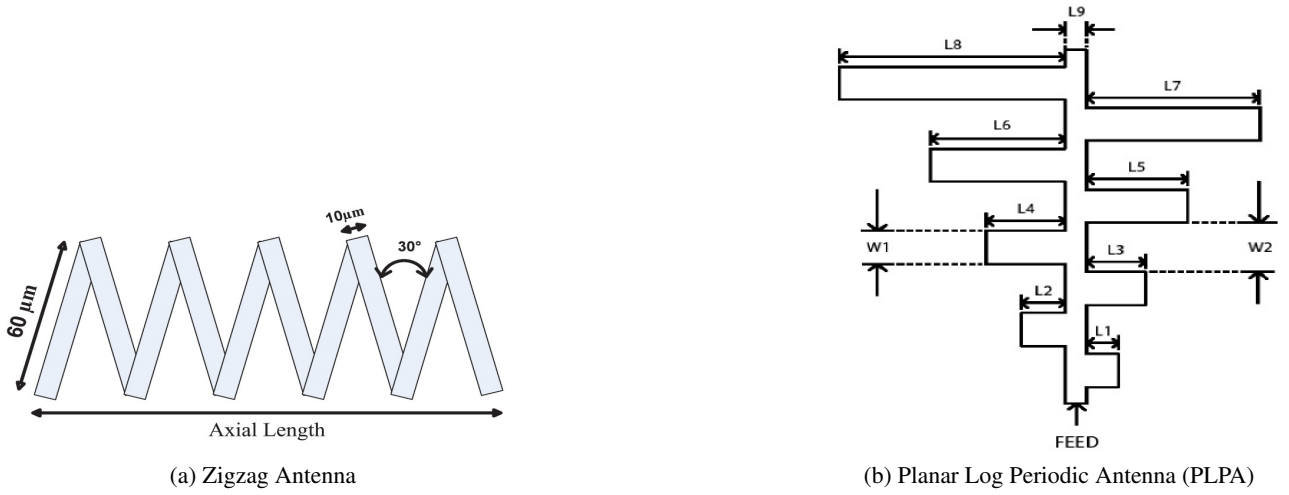


Figure 3.4: Omnidirectional and Directional Antennas in the Proposed Hybrid Wireless Inter-Chip Interconnection

interconnecting GPUs with CPUs over f_{omni} channel. In our implementation, we set this frequency to $45GHz$ and is implemented using Planar Log Periodic Antenna (PLPA)[73]. Since the memory modules communicate over both wireless networks, they are quipped with both antenna types and WIs.

The hybrid inter-chip wireless network for heterogeneous architectures caters to the diverse traffic requirements of both CPU and GPU architectures, while providing low latency, low energy and high bandwidth communication. The directional links provide sustained, high bandwidth memory access for GPU chips, while omnidirectional network provides low latency access between any two chips in the system for efficient remote access. The dual channel wireless operation allows simultaneous, interference free communication to respective memory for both CPU and GPU chips. We have implemented and evaluated the proposed inter-chip interconnection with different modulation techniques and performance capabilities and the necessary wireless infrastructure is described further in detail.

3.4 Inter-Chip Wireless Infrastructure

In this section, we describe the wireless infrastructure to implement the proposed hybrid wireless inter-chip interconnection. It uses omnidirectional and directional antennas to provide the hybrid wireless network for different chips in the system. The dual channel wireless transmission require transceivers operating at f_{omni} and f_{dir} frequencies. Furthermore, we implement the transceivers with two different modulation schemes that provide different bandwidth capabilities. The choice of antennas and transceivers is motivated by the necessity to provide high performance with low area and power overheads.

Table 3.1: Dimensions of PLPA for Directional Wireless Links

Tooth	L1	L2	L3	L4	L5	L6	L7	L8
Size (mm)	0.11000	0.14019	0.17867	0.22772	0.29023	0.36990	0.47144	0.60085
Tooth			L9		W1		W2	
Size (mm)			0.05500		0.11000		0.16500	

3.4.1 Omnidirectional Antenna

The omnidirectional antenna in the hybrid wireless inter-chip network is implemented using Zigzag antenna as shown in Fig. 3.4a. It is designed with $10\mu m$ trace width, $60\mu m$ arm length and 30° bend angle. The axial length of the antenna depends on the operating frequency, f_{omni} , which is designed to operate at $60GHz$ in our implementations. The dimensions of the antenna are shown in Fig. 3.4a. Furthermore, it also has a negligible effect of rotation (relative angle between transmitting and receiving antennas) on received signal strength, making it most suitable for transmission between any two WIs on two remote chips.

3.4.2 Directional Antenna

The directional antenna to form point-to-point wireless links in GPU chips is achieved by use of Planar Log Periodic Antenna (PLPA)[73] as shown in Fig. 3.4b. PLPA is very popular and widely studied for their ease of manufacturing and their wide-band properties. PLPA, shown in Fig. 3.4b, is designed to operate at two frequencies for the requirement GPU's communication with main memory and other chips in the systems as described earlier. The antenna is implemented with eight teeth and the sizes of each teeth increase in logarithmic manner. In our implementation, it is designed to operate at the frequencies of $45GHz$ and $60GHz$ of the hybrid wireless network. The dimensions of the antenna are shown in Table 3.1 with the highest dimension being $1.1825mm$ for the desired frequencies.

3.4.3 Transceiver Circuitry

To ensure high throughput and energy efficiency in proposed hybrid wireless inter-chip network, the transceiver circuitry has to provide wide bandwidth as well as low power consumption. The mm-wave frequency band can provide abundant bandwidth without suffering from severe signal degradation, because of the short communication distances. In our implementation, we have studied two different transceiver circuits with modulation schemes; On-Off Keying (OOK) and Orthogonal Frequency Division Multiplexing (OFDM), which provide

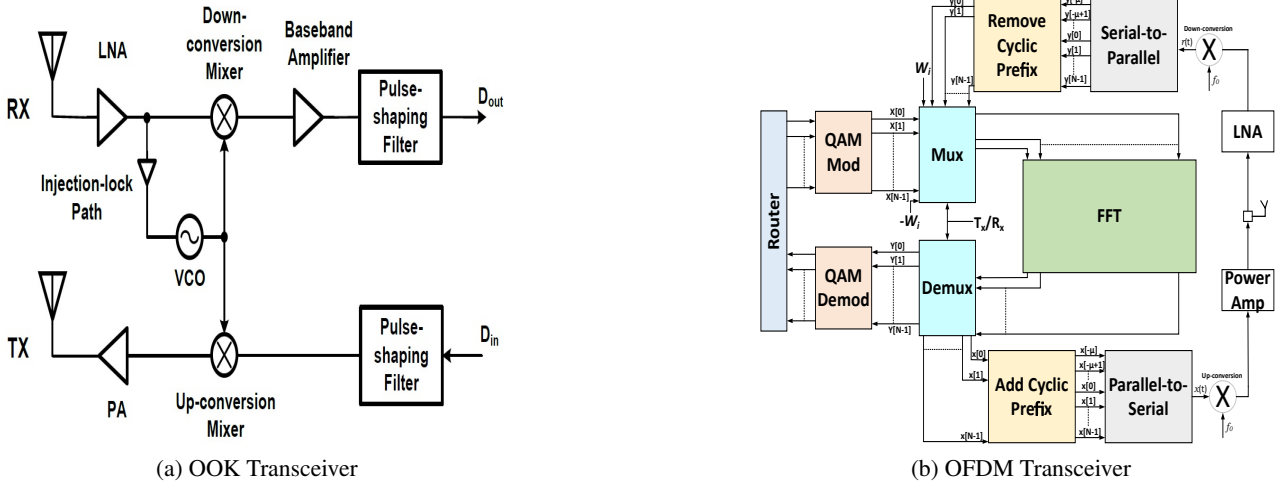


Figure 3.5: Transceiver Circuitry of the Proposed Hybrid Wireless Inter-Chip Interconnection

different bandwidth capabilities.

The transceiver circuit for OOK modulation is shown in Fig. 3.5a and has been adopted from [74]. It implements non-coherence OOK modulation scheme and is chosen for its low overhead implementation. The transmitter (T_X) in OOK transceiver consists of an up-conversion mixer and a power amplifier. On the receiver (R_X) side, direct-conversion topology is adopted, consisting of a low noise amplifier, a down-conversion mixer and a baseband amplifier. An injection-lock voltage-controlled oscillator (VCO) is reused for T_X and R_X to reduce the area overheads. With both direct-conversion and injection-lock technology, a power-hungry phase-lock loop (PLL) is eliminated. The OOK transceiver design keeps the additional overhead added by WIs, while providing sufficient bandwidth for the proposed interconnection.

The OFDM transceiver implemented in the proposed interconnection is shown in Fig. 3.5b. It is broadly divided into three parts; (i) data encoding, (ii) implementation of DFT/inverse DFT and (iii) cyclic prefix and serial/parallel interfaces. We use M-QAM data encoding scheme to ensure high spectral efficiency with sufficient BER. It encodes $\log_2 M$ data bits per symbol into both amplitude and phase of the carrier signal. The FFT module modulates the encoded symbols to respective sub-channel frequency using inverse Discrete Fourier Transform (DFT). The resulting time domain signal is up converted to transmission carrier frequency and amplified using Power Amplifier (PA) before transmission. The FFT radix depends on the number of sub-channels chosen. At the demodulator, received signal is down-converted and converted to parallel data streams. Finally, the cyclic prefix bits and symbols are demodulated using DFT operation of FFT module and decoding operations. The serial/parallel interfaces in the transceiver convert data between parallel data streams to time domain sequence. A detailed discussion of the OFDM design and the factors that impacts its design are discussed in chapter 5.

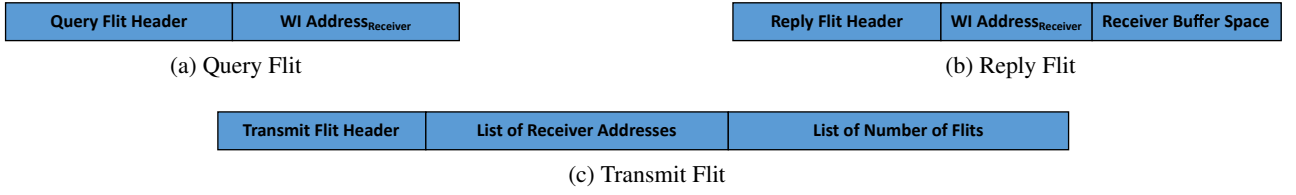


Figure 3.6: Flit Structure for Medium Access Control and Flow Control for Wireless Transmission

3.5 Wireless Communication Protocol

In this section, we describe the inter-chip wireless infrastructure management i.e., medium access control for shared wireless channel and routing algorithm for chip-to-chip communication between different processing and memory chips. We assign each WI across all chips a unique address for medium access control, routing and flow control. The unique WI address, in place of actual destination address, allows for faster address decoding and link establishment, especially in omnidirectional wireless network.

3.5.1 Medium Access Control (MAC)

The dual channel wireless network with omnidirectional and directional links described above enables the simultaneous use of multiple frequency bands and simultaneous transmissions over the point-to-point links. To leverage this, we adopt a hybrid Frequency Division Multiple Access (FDMA) and asynchronous Time Division Multiple Access (aTDMA) based MAC. The frequency bands around wireless channels f_{omni} and f_{dir} are chosen to be non-overlapping to avoid any interference between both wireless networks. The directional wireless network of GPUs and memory establishes point-to-point half duplex links between each memory controller and few respective memory banks. Since no two directional wireless links have overlapping radiation regions, medium access control for directional wireless network is not required. For any memory access on a GPU chip, the request is routed to appropriate memory controller on the chip and then transmitted over the wireless link. While all memory banks linked to this memory controller receive the request, only the memory bank with required data responds to the request over directional link. This provides the dedicated memory access bandwidth necessary for the GPUs without any stalling.

The non-directional antennas operating at f_{omni} band, deployed in multicore CPUs to enable communication with any other system component, can be oriented in any direction with respect to itself. Due to the potentially large number of WIs across the system in this band, an efficient collision avoidance MAC is necessary. We adopt aTDMA approach with Carrier Sense Multiple Access with Collision Detection (CSMA/CD) for establishing wireless links on-demand. In this MAC mechanism, all WIs sense the omnidirectional wireless

channel for any ongoing transmission. The flit structures are shown in Fig. 3.6. When the channel is sensed to be free and a WI has data to send, it transmits a short *query flit* with only WI address of desired destination as shown in Fig. 3.6a. As the *query flit* reaches desired WI, it copies the *query flit*'s content (i.e. the added WI address) and broadcasts a *reply flit* containing this address and the available buffer space in the destination WI as shown in Fig. 3.6b. The buffer space field indicates the available free space at the input buffers of the receiving WI. If the available buffer space is not sufficient to store the receiving data, the transmission is delayed as described in flow control. As the sender WI receives the reply flit, it broadcasts a *transmit flit*, shown in Fig. 3.6c, with the number of flits to be transmitted. Both the *query flit* and *transmit flit* are received by all WIs, but only the destination WI keeps its receiver ON for the time period mentioned in the header flit of *transmit flit* packet. Other WIs start a wake-up timer initialized with the value provided in the *transmit flit* and go to sleep, which helps maintain energy efficiency with CSMA MAC. After the timer expires, all WIs wake up and go back to sensing mode.

A collision is detected if there is a failure in receiving *reply flit* at sender WI within a specific time after the *query flit* is sent. In case of a collision, the colliding WIs back off from transmitting through wireless channel for time B_T given by,

$$B_T = r * \eta_B * t_{flit} \quad (3.1)$$

where η_B is the number of flits a WI must wait before trying to access the channel, t_{flit} is the time to transmit a flit over the wireless medium and r is a back-off period, which is the duration of back-off before the next transmission is attempted by the WI. This pseudo-random number is generated within a contention window generated by multiplying the back-off factor with a uniform random integer generated by another LFSR implementing exponential back-off. Such back-off, reduces the probability of the WIs from retransmitting at the same time after a collision and reduces probability of collision in future transmissions[75]. When a WI enters the back-off state, it can only receive flits transmitted through the wireless medium. In the back-off period, if a WI receives query flit and transmit flit, back-off timer is paused till the wake-up timer expires. The back-off timer restarts counting as the wake-up timer expires.

3.5.2 Flow Control and Routing

In proposed inter-chip wireless interconnection, the end-to-end routing and flow control is dependent on the final destination of the packet. The intra-chip networks in both CPU and GPU chips, based on wired mesh topology, use *XY* routing algorithm with wormhole flow control for packet transfers. When both source and destination nodes are within the same chip, data is routed simply using the *XY* routing algorithm. When the

source and destination nodes are from different chips, data from the source is first routed to the appropriate WI using *XY* routing. The appropriate WI in the source chip depends on the type of packet and type of chip. In CPU chips, it is the nearest WI as each WI can establish communication with any the desired destination, processing or memory chip. In case of GPUs, the choice of WI depends on whether the packet is either memory bound or processor bound. For former, the packet is routed to the WI which establishes the directional link to the memory bank with required data. For the latter, the packet is routed to the nearest WI for establishing wireless transmission with desired destination processor, which can be CPU or another GPU chip.

Once the packet reaches the appropriate WI, it requests access for the wireless channel with destination being the address of the WI of the packet destination chip. All inter-chip communication, except for GPU to memory communication, is done over omnidirectional network, while the GPU to memory communication uses the directional links. Hence, on GPU and memory chips, the WI makes the decision to use f_{omni} or f_{dir} channel depending on the packet destination. As described before, any memory request from/to GPU chip does not require request for channel resources as it is transmitted using directional wireless links over f_{dir} channel. For all other communication from/to CPU/GPU over f_{omni} frequency, the channel access mechanism for the same is described in the previous section. While establishing wireless link on-demand in the omnidirectional network, if the buffer space in the *reply flit* is smaller than the number of flits to be transmitted, the transmitter waits a predefined time before initiating the data transmission. This waiting time is set depending on the average waiting time at any WI for routing packets from its input buffer to output of the router at which the WI is attached. Once channel access is obtained for packet transmission, the packet is transmitted to destination WI. The data from receiving WI is routed to final destination using *XY* routing algorithm in destination chip. In the scope of this work, we assume that the flit sizes on all chips is same, though this is not necessary for the proposed interconnection.

3.6 Experimental Evaluation

In this section, we discuss the experimental setup, architecture of baseline and proposed systems evaluated and performance improvements provided by proposed design.

3.6.1 Experimental Setup

To evaluate proposed inter-chip wireless architecture for CPU/GPU heterogeneous systems, we use Multi2Sim [36] simulator. It is a cycle accurate heterogeneous architectural simulator, that allows detailed configuration of

Table 3.2: Details of CPU/GPU Heterogeneous Systems

Chip	Component	Configuration
CPU	Core	x86 cores, 18 cores/chip, 2.5 GHz
	L1 Cache	8-way, 64 KB cache, 4 cycle, LRU policy, 64 B line, private
	L2 Cache	4-way, 256 KB cache, 12 cycle, LRU policy, 64 B line, private
	L3 Cache	16-way, 2 MB/core cache, 42 cycle, LRU policy, 64 B line, shared
GPU	Core	Southern Islands, 32 compute units, 64 stream cores/compute unit, 1.5 GHz
	L1 Cache	8-way, 32 KB cache, 4 cycle, LRU policy, 64 B line, private
	L2 Cache	8-way, 1 MB/core cache, 10 cycle, LRU policy, 64 B line, shared

ISA, memory hierarchy and interconnection topology. The architectural configuration and memory hierarchy of both CPU and GPU chips used in our evaluation is shown in Table 3.2. The CPU chips are modelled after contemporary high-performance processors like Intel Xeon[70] and AMD EPYC[50]. Each CPU is comprised of 18 cores and each GPU is comprised of 32 compute units. Each CPU core has private L1 and L2 caches, while L3 cache is shared by all cores. The L3 cache is divided physically into multiple banks (equal to number of cores) and each chip has a total of 36 MB of L3 cache. The CPU core, its private caches and L3 cache segment, along with wired router form a single node as shown in Fig. 3.3. In GPUs, each compute unit has a private L1 cache, while second level cache is shared by all compute units. We evaluate performance and overheads for two systems; (i) 2 CPU and 2 GPU chips (2C-2G) and (ii) 4 CPU and 4 GPU chips (4C-4G).

The configuration of proposed inter-chip interconnection using wireless links for heterogeneous multi-chip systems is shown in Table 3.3. The intra chip network, in both baseline and proposed architectures, is implemented using wired mesh topology. The router and link configuration are shown in Table 3.3. The intra chip routers operate at the same frequency as that of the cores in respective processing chips. The inter-chip interconnection for proposed architecture is implemented using hybrid wireless design with omnidirectional and directional links. The communication between a CPU and all other system chips viz., GPU, CPU and main memory is transmitted using an omni-directional wireless link operating at 60 GHz carrier frequency. The GPU to main memory traffic is transmitted over directional point-to-point wireless links implemented using PLPA with 45 GHz carrier frequency. Each chip is fitted with wireless interfaces for all inter-chip communication as described in previous sections. The wireless links are implemented using two different modulation techniques, OOK and OFDM as described. The specifications of both designs is provided in Table 3.3. The OOK modulation provides a bandwidth of 16Gbps per link with 1.2pJ/bit energy consumption, while each OFDM wireless link provides 195.32Gbps bandwidth at 0.132pJ/bit energy consumption. The 2C-2G and 4C-4G systems,

Table 3.3: Details of Intra and Inter Chip Network

Component	Configuration
Intra Chip Network	
Topology	Wired Mesh for both CPU and GPU
Routers	5 ports (including local port), 64-bit flit, 8 flits buffer per port
Links	64 bits wide, 1 cycle latency
Inter Chip Network	
CPU-CPU	Omni-directional wireless link, 60 GHz carrier
CPU-GPU	
CPU-Memory	
GPU-Memory	Directional wireless link, 45 GHz carrier
OOK Wireless	16 Gbps/link, 1.2 pJ/bit
OFDM Wireless	195.32 Gbps/link, 0.132 pJ/bit

Table 3.4: Total Bandwidth of the Proposed Inter-Chip Wireless Interconnection

System	Number of Links	OOK	OFDM
2C-2G	9	144 Gbps	1.72 Tbps
4C-4G	17	272 Gbps	3.24 Tbps

evaluated in this work, have a total of 9 and 17 wireless links operating simultaneously and the total inter-chip bandwidth provided by proposed interconnection is shown in Table 3.4. As observed, the proposed inter-chip wireless interconnection provides multiple terabits per second bandwidth using OFDM links. Prudent link, antenna and transceiver designs have the potential to provide significantly higher bandwidths.

The efficiency of proposed inter-chip interconnection is evaluated by simulating different CPU and GPU applications from PARSEC[76], SPLASH2[77], AMD APP SDK[78] and Rodinia[79] benchmark suites, shown in Table 3.5. Application scenarios are chosen to be CPU only, GPU only and concurrent execution of both CPU and GPU applications to sufficiently stress the system. In all scenarios, problem sizes are chosen such that all cores in CPU and/or GPU are active for the entire duration of execution. This is to ensure that the system is properly stressed and we capture contention for off-chip resources also. We evaluate network energy, network latency and application runtime of proposed topology and compare it against those of baseline system.

Table 3.5: CPU and GPU Applications Evaluated

CPU	Barnes (BN), BlackScholes (BK), FFT, LU, Ocean (OC), Radiosity (RT), Radix (RX), Ray-trace (RY), Swaptions (SW), X264
GPU	Binary Search (BS), Back Propagation (BP), Convolution (CL), DCT, Eigen Value (EV), Fast Walsh (FW), Histogram (HG), Matrix Multiplication (MM), Nearest Neighbour (NN), Quasi Random (QR), Radix Sort (RS), Stream Cluster (SC)

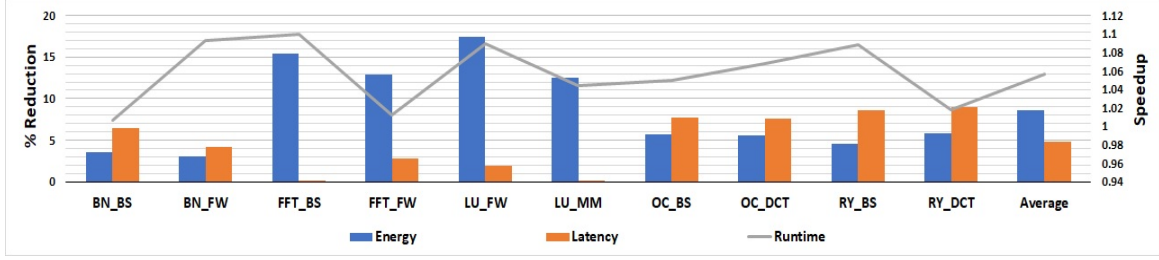


Figure 3.7: Comparison between hybrid wireless interconnection and omnidirectional only wireless network

3.6.2 Hybrid Wireless Interconnection

In this section, we briefly evaluate the motivation for implementing hybrid wireless interconnection with directional and omnidirectional antennas. Fig. 3.7 shows the improvements provided by hybrid wireless interconnection as compared with inter-chip wireless network with only omnidirectional antennas on a system with two CPUs and two GPUs. The results are obtained using OOK modulation described in Table 3.3 and the system with only omnidirectional links is implemented as a dual channel wireless network with $45GHz$ and $60GHz$ carriers. As can be observed, the hybrid wireless interconnection provides better energy savings upto 20% and runtime speedup upto $1.1\times$ as compared to using only omnidirectional antennas. This improvement is primarily due to reduced contention for wireless channel and interference free communication for GPUs. While a system with only directional wireless links may reduce interference for inter-chip interconnection and provide higher overall throughput, it poses implementation challenges. It is infeasible to implement several directional wireless links between all chips without any overlap between their radiation regions. Furthermore, directional antennas incur higher area overheads as compared to omnidirectional antennas as detailed in section 3.6.5.

3.6.3 Network Energy

One of the major advantages of wireless interconnects is their energy efficiency over wired links for long distance communication. Fig. 3.8 and Fig. 3.9 show the saving in average packet energy in the network using inter chip wireless interconnection as compared to baseline wired interconnection for both system sizes evaluated.

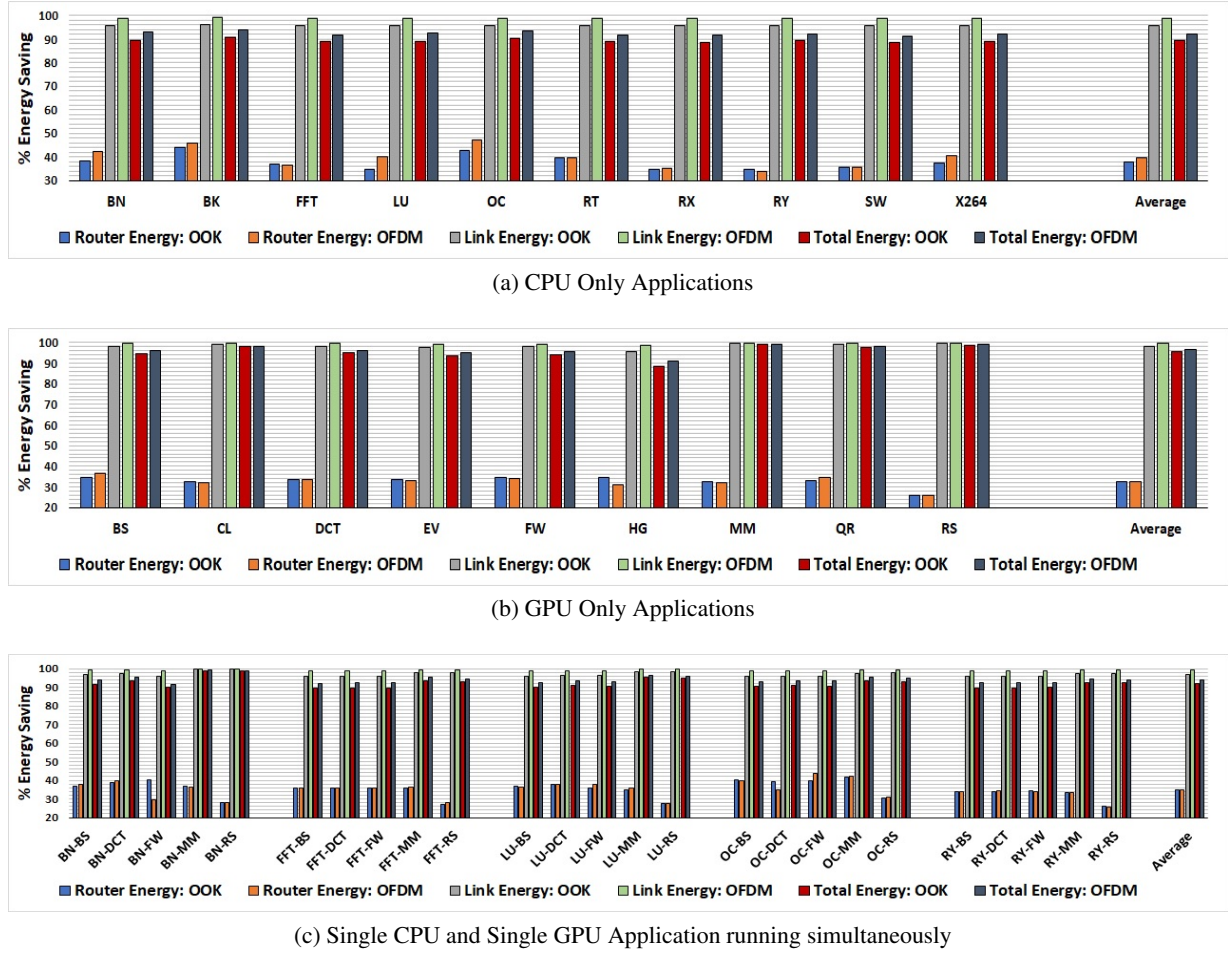


Figure 3.8: Average packet energy saving achieved by wireless inter chip interconnection against traditional wired network on system with two CPU and two GPU chips

The energy results are shown for both OOK and OFDM based implementations. The results also show savings in routers and links separately to clearly highlight the contribution of wireless links over wired links. The interconnect energy includes both on-chip and off-chip packet communication, along with serialization, antenna and analog component overheads of the wireless interface. These overheads are discussed in detail in section 3.6.5. As observed from Fig. 3.8, with single CPU and single GPU application running simultaneously, OOK and OFDM based inter-chip wireless network saves 92.08% and 93.409% of average packet energy respectively as compared to baseline interconnection on a system with two CPU and two GPU chips. Similarly, on system with four CPU and four GPU chips from Fig. 3.9, OOK and OFDM based inter-chip wireless interconnection saves packet energy of 92.94% and 93.46% respectively as compared to baseline. Furthermore, the improvement in network energy is higher when multiple applications are running simultaneously as opposed to single application, showing the scalability of wireless interconnection under high stress.

Fig. 3.10 shows the contribution of different on-chip and off-chip components to the overall network

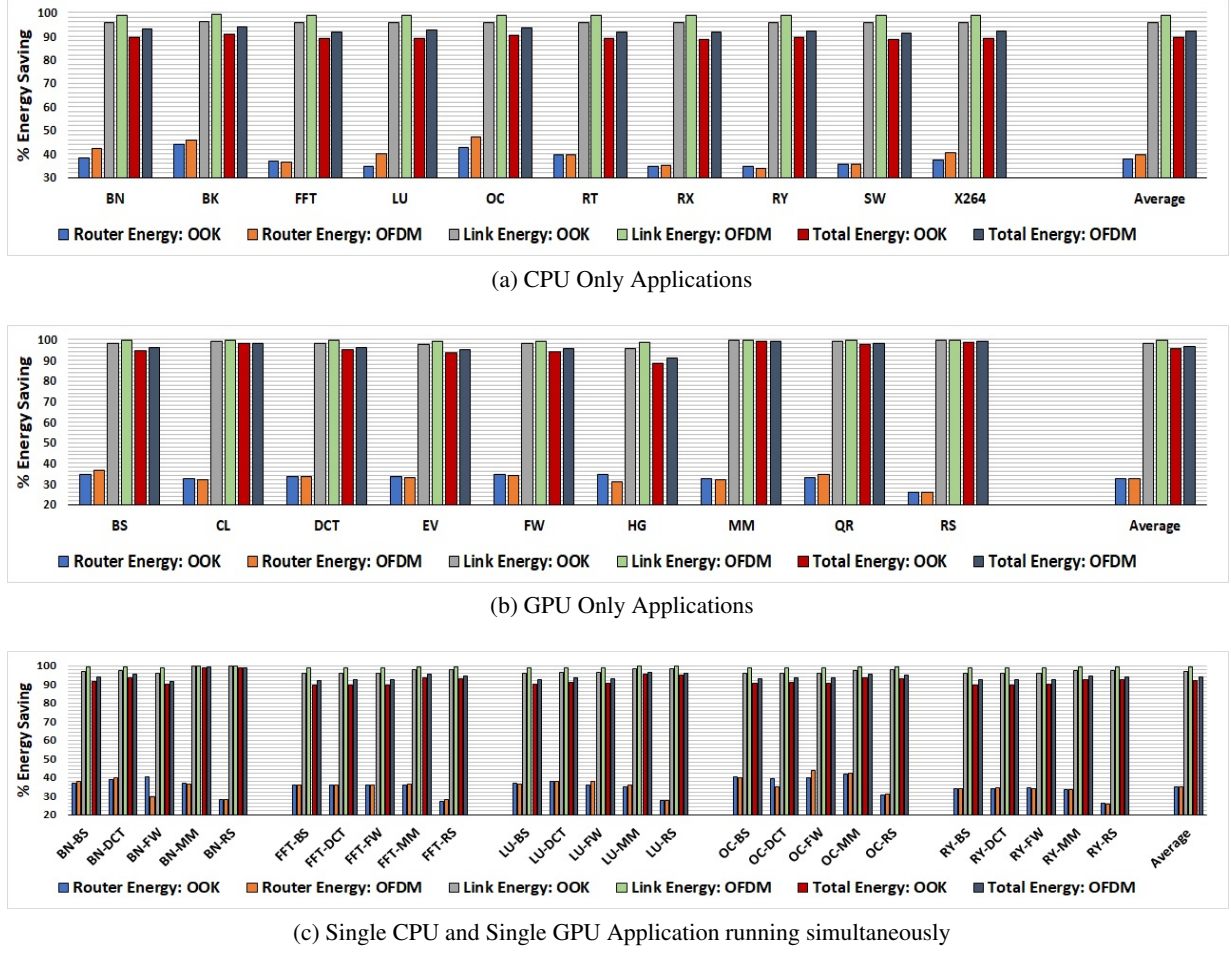
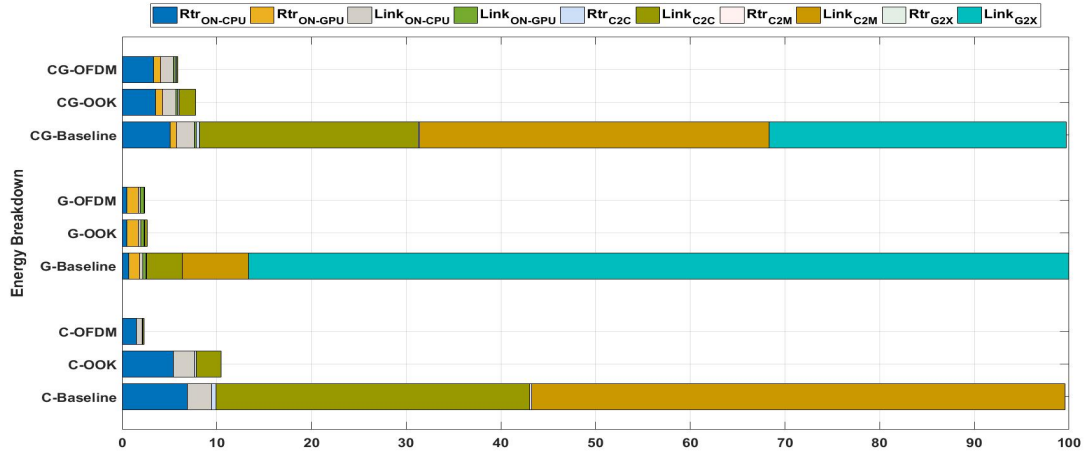
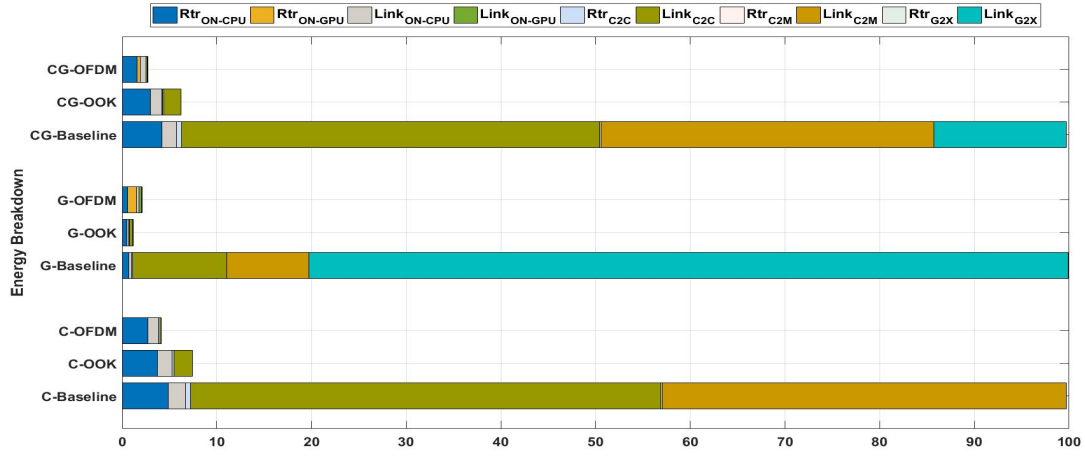


Figure 3.9: Average packet energy saving achieved by wireless inter chip interconnection against traditional wired network on system with four CPU and four GPU chips

energy consumption under different application scenarios. The energy breakdown results are averaged over all benchmarks in each application scenario; CPU only (C), GPU only (G) and CPU+GPU (CG) for baseline, OOK based and OFDM based wireless interconnections. In Fig. 3.10, Rtr_{ON-CPU} and Rtr_{ON-GPU} represent router energy within CPU and GPU chips respectively, while $Link_{ON-CPU}$ and $Link_{ON-GPU}$ represent link energy within CPU and GPU chips respectively. The off-chip network energy consumption is broken down in CPU-to-CPU energy (Rtr_{C2C} and $Link_{C2C}$), CPU-to-Memory energy (Rtr_{C2M} and $Link_{C2M}$) and GPU-to-Any energy (Rtr_{G2X} and $Link_{G2X}$) components. In the baseline system, the off-chip wired links contribute to most of the network energy consumption, accounting for 92.39% and 94.03% of total network energy on 2C-2G and 4C-4G systems respectively across different application scenarios. In baseline architecture, all GPU off-chip communication happens over PCIe bus, while CPU off-chip communication happens over memory and chip-to-chip buses respectively. In comparison, the OOK based wireless inter-chip interconnection reduces the contribution of off-chip link energy to 30.49% and 33.80% of total network energy on 2C-2G and 4C-4G systems respectively. In the wireless inter-chip interconnection, $Link_{C2C}$ represents all communication over



(a) 2C-2G System



(b) 4C-4G System

Figure 3.10: Breakdown of network energy consumption on baseline and wireless inter-chip interconnection for different application scenarios

the omnidirectional network, while $Link_{G2X}$ represents directional wireless network communication. The off-chip link energy consumption is further reduced to 10.85% and 10.51% of total network energy using OFDM based wireless links because of their better per bit energy efficiency.

As observed from Fig. 3.10, the network energy saving achieved with wireless inter-chip interconnection clearly highlights the efficiency of wireless links in optimizing energy consumption for off-chip communication as compared to PICE and bus based interconnection of baseline architecture. Using a hybrid combination of such energy efficient wireless links, the proposed inter-chip interconnection reduces link energy by 97.39% and 99.16% with OOK and OFDM modulation respectively on the 4C-4G system as seen from Fig. 3.9. The wireless links, besides exhibiting better energy efficiency as compared to wired links, also reduce application

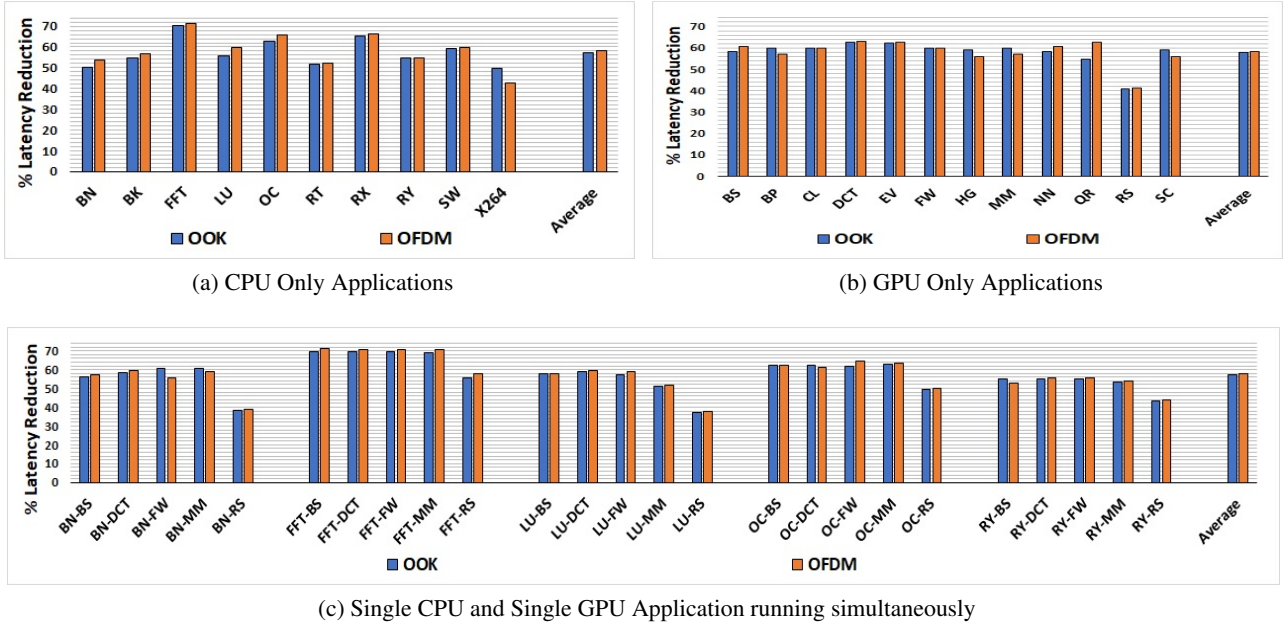


Figure 3.11: Reduction in average packet latency using wireless inter chip interconnection against traditional wired network on system with two CPU and two GPU chips

runtime and overall active period of both on-chip and off-chip network components. Combined with lower average hop count to reach off-chip switches, the wireless interconnection also saves router energy by 38.21% on the 4C-4G system, further improving the network energy efficiency as compared to wired interconnection of baseline architecture.

3.6.4 Performance Evaluation

The performance of the proposed inter-chip wireless interconnection is evaluated in terms of average packet latency of the network and total application runtime.

3.6.4.1 Network Latency

One of the major challenges for performance in many core architectures is off-chip memory access latency and bandwidth. This is further exacerbated in multi-chip systems, especially for accessing remote chip and memory elements. Wireless links exhibit linear latency characteristics over large distances and provide high bandwidth making them ideal for overcoming inter-chip performance bottlenecks. Fig. 3.11 and Fig. 3.12 show the reduction in average packet latency using inter-chip wireless interconnection in comparison with baseline system using wired bus and PCIe links on 2C-2G and 4C-4G systems respectively. On average,

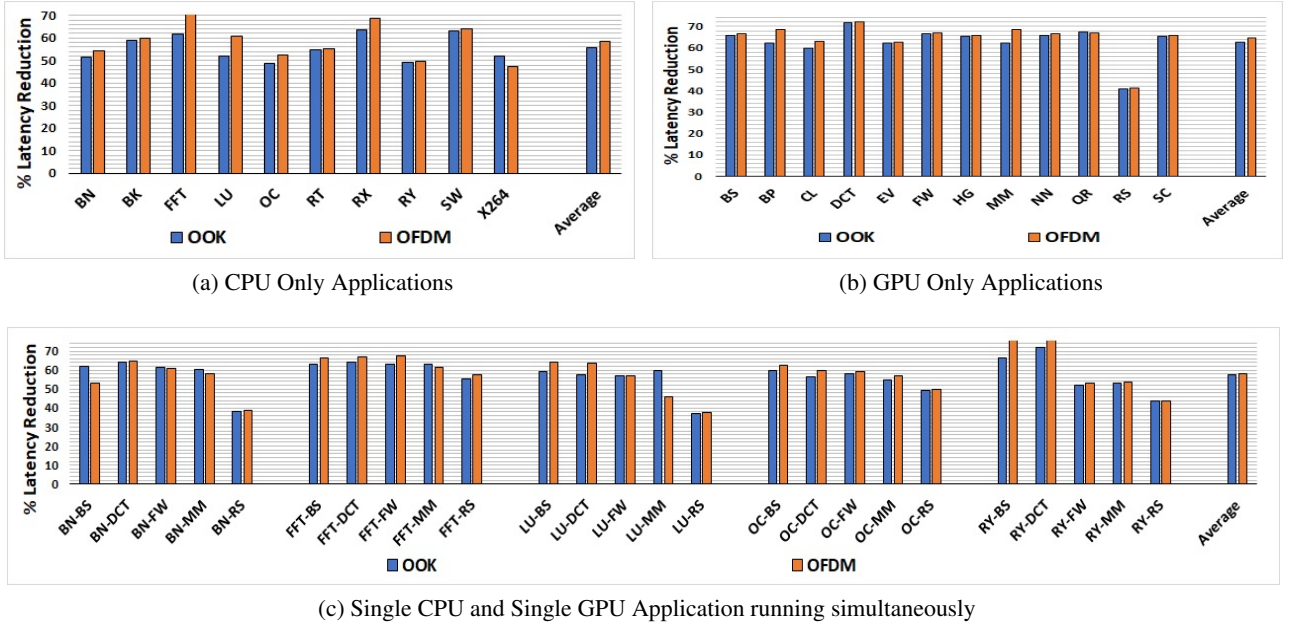
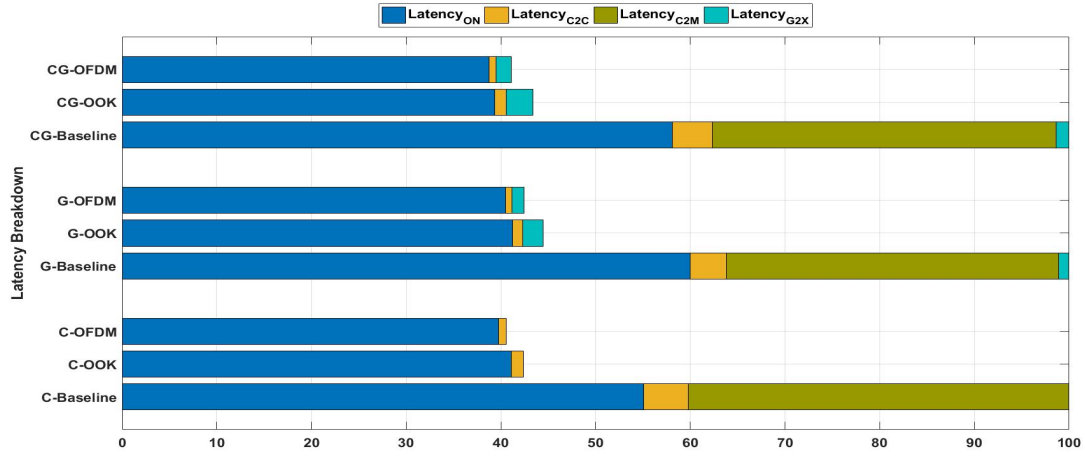


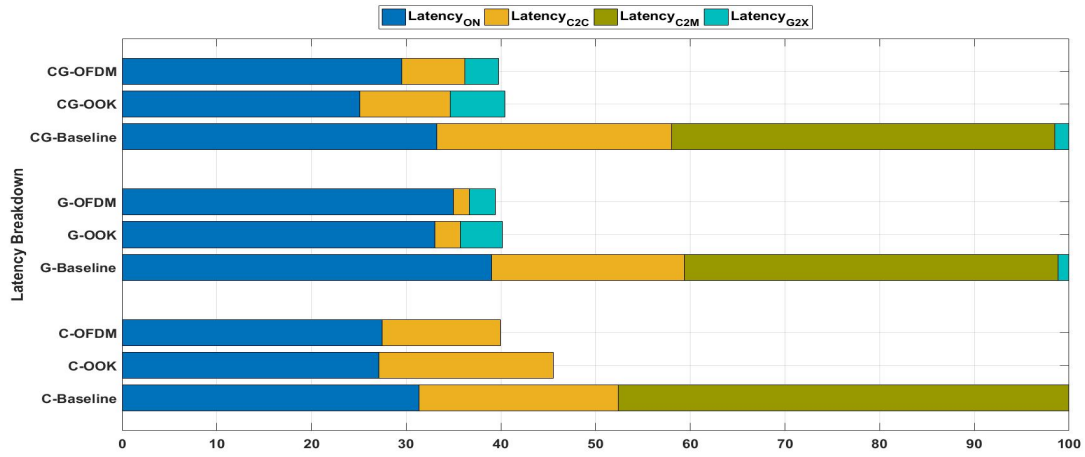
Figure 3.12: Reduction in average packet latency using wireless inter chip interconnection against traditional wired network on system with four CPU and four GPU chips

across different concurrent application scenarios, the packet latency is reduced by 57.19% and 57.32% for 2C-2G and 4C-4G systems respectively using OOK based inter-chip wireless network as compared to baseline interconnection. Using the OFDM based wireless links, the average packet latency is reduced by 57.59% and 58.34% for 2C-2G and 4C-4G systems respectively. The network performance scales with multiple applications running simultaneously as observed. Furthermore, wireless links with their superior performance across long distances avoid the limitation of keeping off-chip I/O components along chip edges as is traditionally done. This provides freedom to designers to optimally place the off-chip switches across the chip to optimize intra chip performance for off-chip communication and also reduce on-chip average packet latency.

To further highlight the efficiency of wireless links in optimizing off-chip communication performance, Fig. 3.13 shows the breakdown of total packet latency into different latency components under different application scenarios for all evaluated architectures. Similar to energy breakdown, the latency breakdown is also shown for each application scenario for baseline, OOK based and OFDM based wireless interconnections. $Latency_{ON}$ in Fig. 3.13 represents the average packet latency within chip components (both CPU and GPU), while $Latency_{C2C}$, $Latency_{C2M}$ and $Latency_{G2X}$ represent the off-chip latency for CPU-to-CPU, CPU-to-Memory and GPU-to-Any communication respectively. Fig. 3.13 shows that the off-chip communication latency contributes to a significant portion of the overall packet latency in baseline architecture under any application scenario. On average, off-chip communication latency accounts for 42.66% of total latency under different application scenarios on 2C-2G system as seen from Fig. 3.13a. The major portion of off-chip



(a) 2C-2G System



(b) 4C-4G System

Figure 3.13: Breakdown of average packet latency on baseline and wireless inter-chip interconnection for different application scenarios

communication latency is contributed by main memory communication to service all memory requests from CPU/GPU. The contribution of off-chip latency increases to 66.05% on the 4C-4G system as seen from Fig. 3.13b. Especially, the $Latency_{C2C}$ increases from $\approx 4\%$ on 2C-2G system to $\approx 23\%$ on 4C-4G system clearly highlighting the limitations of wired inter-chip interconnection in servicing remote requests across different chips, along with main memory communication.

In contrast, the OOK and OFDM based wireless inter-chip interconnection reduces the off-chip communication latency to 5.56% and 3.58% of total latency respectively on 2C-2G system as seen from Fig. 3.13a. In the figure, $Latency_{C2C}$ represents omnidirectional wireless network latency and $Latency_{G2X}$ represents directional network latency. On 4C-4G system from Fig. 3.13b, the OOK and OFDM based off-chip wireless

links contribute to 8.38% and 5.08% of total network latency under different application scenarios. The slight increase in the off-chip communication latency on 4C-4G system is attributed to the increased waiting latency at each transceiver for accessing omnidirectional channel. From Fig. 3.13, the low latency communication of wireless links across long distances clearly improves off-chip communication significantly in accessing both main memory and remote chips. This is further aided by the hybrid interconnection, which minimizes waiting and channel access latencies for GPU off-chip communication and improves overall off-chip communication performance. The off-chip wireless communication latency, as number of chips in the system increases, can be optimized by exploring and adopting efficient channel sharing[80] and channel access mechanisms[81] to reduce waiting time at each transceiver. Furthermore, the on-chip communication latency can also be improved by optimizing the placement of I/O and transceivers across the chip as opposed to chip edges as is the case in wired inter-chip interconnection architecture.

3.6.4.2 Application Runtime

The application runtime improvements using wireless interconnection over baseline are quantified by speedup in execution time for completing application contexts (both CPU threads and GPU kernel (if present)). An application is comprised of both computation and memory instructions and speedup is dependent not only on number of memory accesses, but also on distribution of memory accesses throughout the application execu-

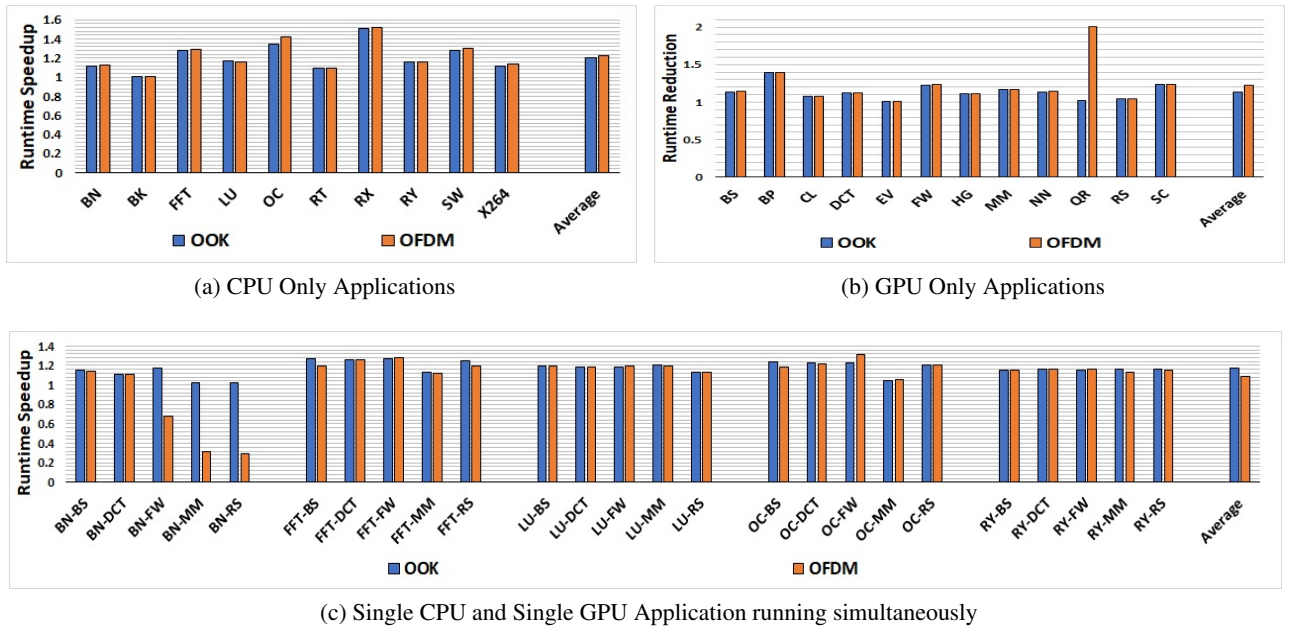


Figure 3.14: Application runtime speedup using wireless inter chip interconnection against traditional wired network on system with two CPU and two GPU chips

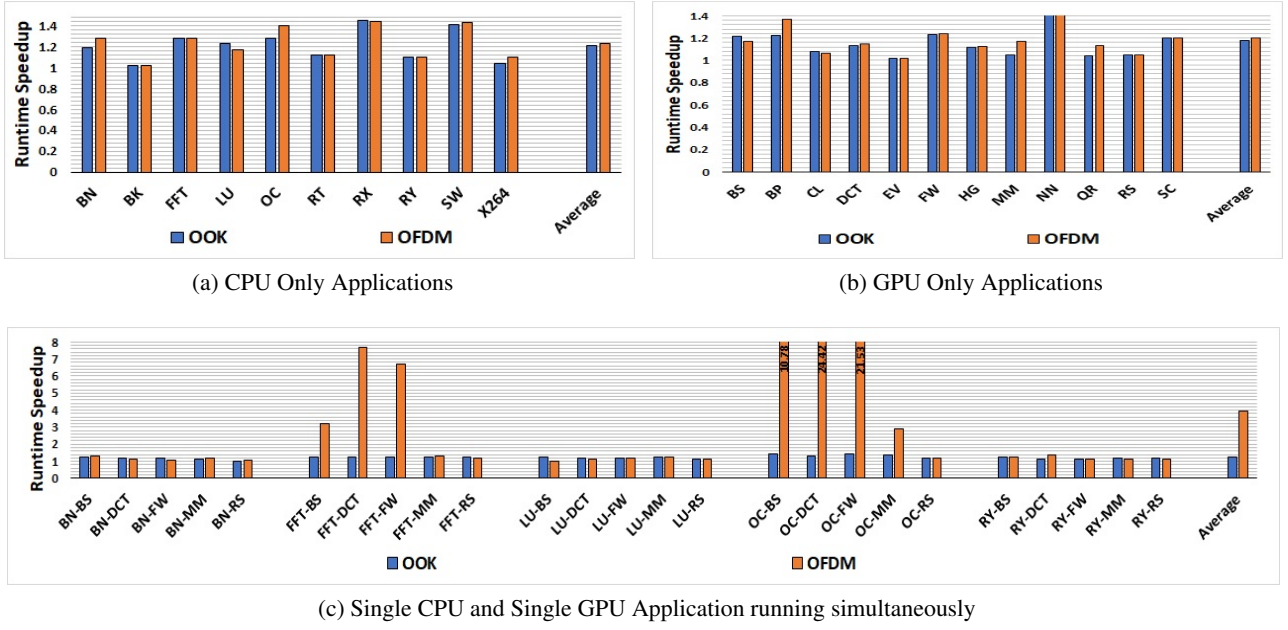


Figure 3.15: Application runtime speedup using wireless inter chip interconnection against traditional wired network on system with four CPU and four GPU chips

tion. The application speedup with wireless interconnection over wired interconnection for CPU, GPU and CPU+GPU applications in both system sizes is shown in Fig. 3.14 and Fig. 3.15. Figures 3.14a, 3.15a and Figures 3.14b, 3.15b show runtime speedup when only either a CPU or GPU application is running on the system. In such scenarios, the speedup in application is solely due to faster access to off-chip memory and remote locations in other chip components. In case of GPU only application scenarios, a section of application runs on CPU which handles mainly control tasks, while computation is performed by GPU kernel. Even with baseline system using dedicated memory buses for each memory controller, wireless inter-chip interconnection using OFDM wireless links improves application runtime by $1.22\times$ and $1.24\times$ in CPU only application scenarios on 2C-2G and 4C-4G systems respectively. This clearly shows the performance improvements provided by inter chip wireless interconnection, even with all CPU chips sharing single channel resources. This is due to significantly high bandwidth and low latency provided wireless links as compared to off-chip buses. In GPU only application scenarios, Fig. 3.14b and Fig. 3.15b show that the runtime is sped up by $1.10\times$ using OOK based wireless links as compared to PCIe based interconnection. With OFDM based wireless links, the runtime is further improved by $1.22\times$ and $1.12\times$ on both systems respectively. Even when the system is stressed with concurrent CPU and GPU applications and channel resources being shared, application runtime is improved by $1.17\times$ and $1.23\times$ for 2C-2G and 4C-4G systems respectively using OOK based wireless links as compared to baseline interconnection. Using OFDM wireless links, the application runtime is further improved to $1.18\times$ and $3.94\times$ on 2C-2G and 4C-4G systems respectively in comparison with baseline interconnection as shown

Table 3.6: Area and Power Overheads of Inter-Chip Wireless Interconnection

Component	Wired Router	OOK WI	OFDM WI	Zigzag Antenna	PLPA
Area (mm^2)	7.67×10^{-3}	0.30	0.15	0.022	1.33
Power (mW)	4.46	28	25.825	–	–

in Fig. 3.14c and Fig. 3.15c. The improvements in network latency and application runtime clearly show the potential of hybrid wireless inter-chip interconnection in providing efficient communication performance in heterogeneous multichip systems. The performance can be further improved by designing suitable protocol infrastructure like scheduling algorithms, memory access mechanisms that compliment wireless interconnection.

3.6.5 Overhead Analysis

In this section, we evaluate the area and power overheads required to implement the proposed inter chip wireless interconnection. All wired routers are synthesized using $32nm$ technology, while wireless transceiver components are synthesized at $45nm$. Table 3.6 details the overheads of both intra and inter-chip networks of the proposed interconnection. The wired components of the intra chip network in both CPU and GPU are designed and synthesized in Synopsys Design Compiler. Each wired router is comprised of 5 ports, including the local port and is input buffered with 8-flit wide buffers. It occupies a total area of $7.67 \times 10^{-3}mm^2$ including all the components. The proposed inter-chip wireless interconnection replaces all I/O controllers in baseline system with wireless interfaces for inter-chip communication. As seen from Fig. 3.2 and Fig. 3.3, it replaces the six wired I/O (including memory controllers) in CPU chips and eight I/O in GPU chips of baseline system with four WIs and antennas each. The WI area overhead is the total area consumed by modulator/demodulator, power amplifier and LNA circuits. The circuitry for OOK modulation requires a total area of $0.16mm^2$, while OFDM circuit with 256 sub-channels occupies an area overhead of $0.092mm^2$ for each transceiver. The power amplifier and INA circuits in both OOK and OFDM WIs add an area overhead of $0.14mm^2$. Hence, the total area overhead for each WI is $0.30mm^2$ and $0.15mm^2$ for OOK and OFDM modulations respectively. In addition, the omnidirectional zigzag antenna is $0.38mm$ long and $58\mu m$ wide (area of $0.022mm^2$), while PLPA for directional links has an area of $1.33mm^2$.

Each wired router in the intra chip network consumes total dynamic and static power of $4.46mW$ and $6.44mW$ respectively. In WI, the power amplifier and LNA contribute to the majority of its power consumption. Together, they both consume $19mW$ of power. The circuitry for OOK modulation consumes $9mW$, while OFDM with 256 sub-channels has a power overhead of $6.825mW$. In total, each WI consumes $28mW$ and

25.825mW for OOK and OFDM modulation respectively. The network energy results presented in the previous section accounts for all WI overheads at each WI. These power overheads, especially static and idle state power consumption, can be minimized by low power techniques[82, 83] to further reduce the network energy consumption.

3.7 Extension to Any Heterogeneous Architecture

The proposed hybrid inter-chip wireless interconnection shows tremendous improvements in performance and energy efficiency for heterogeneous multichip architectures. While the interconnection design has been discussed in the context of CPU/GPU multichip systems, it is extensible to any heterogeneous multichip architecture. By abstracting the major characteristics and requirements of all inter-chip communication, a prudent and efficient hybrid wireless interconnection can be designed for a heterogeneous system with any number of different architectures.

A typical heterogeneous multichip system consists of CPU acting as master with other application specific hardware for improving performance and/or efficiency. Besides communication with CPU and main memory, the important off-chip and inter-chip communication behaviours that impact the design of wireless interconnection are:

- **Computation/Memory Boundedness:** One of the basic and important constraints impacting off-chip interconnection design is the memory or computation boundedness of the applications running on specialized hardware. While computationally intensive applications are less impacted by inter-chip interconnection, memory intensive applications require efficient interconnection architecture. Especially with rise of memory bound applications like machine learning, artificial intelligence, etc., data movement from/to main memory and other chips (if any) contributes most to the application execution. Hence, a bandwidth/energy efficient interconnection design for such accelerators in the heterogeneous system is a necessity.
- **Latency/Bandwidth Sensitivity:** Similar to computation or memory intensive nature of applications, the latency or bandwidth sensitivity of an architecture imposes the primary constraints for off-chip (and also on-chip) interconnection design. The latency/bandwidth sensitivity of an architecture is determined by factors like predominantly serial/parallel execution at each core, spatial and temporal locality of memory accesses and instruction scheduling techniques, etc. While most architectures fall into either category, significant challenges arise to meet conflicting requirements on shared off-chip interconnection, when

architectures of different sensitivities are integrated in the heterogeneous system.

- **Temporal Nature of Off-Chip Memory Accesses:** The temporal behaviour of off-chip memory accesses, determined by the average number of memory accesses per compute, is dependent on application, computing architecture, reuse of cached data, etc. and significantly impacts off-chip communication. Architectures with low volume, sporadically distributed off-chip memory accesses tolerate longer off-chip communication latencies and waiting time for resources through efficient instruction scheduling and data prefetching. On the other hand, dense temporal distribution or high volume of sparsely distributed memory accesses severely impact performance under long off-chip latency and interference from other inter-chip communication. Such architectures require prudent and efficient off-chip interconnection design to avoid high latency and energy consumption, interference for shared off-chip resources, etc. to achieve optimal performance.
- **Data sharing among chips:** Prevalence of data sharing among multiple chips require an inter-chip interconnection that can enable transmission of requests and updates to shared data across all chips with low latency and low energy. Additionally, existence of communication with multiple chips require sharing of off-chip interconnection and leads to interference and increased communication latency. Hence, an interconnection that balances resource sharing and inter-chip communication performance is required.
- **Real Time Constraints:** In addition to achieving average performance, communication constraints in real time applications are characterized by bounded and worst case performance to avoid any system failures. Traditional interconnection design focuses on optimizing average performance and has the best and worst case performance as secondary constraints. System design and consequently interconnection design for real time architectures takes the opposite direction and further complicates inter-chip interconnection design when integrated with other architectures in the heterogeneous systems. Hence, additional constraints have to be imposed to meet worst case performance specifications, while optimizing average interconnect performance. These constraints can be in the form of minimizing sharing of off-chip interconnection resources, interference from other chips, channel access latency, etc. in inter-chip wireless network take precedence to meet the real time constraints.

In addition to these salient communication characteristics described above, application dependent behaviours like data reuse and memory subsystem specifications also impact the design of inter-chip wireless interconnection in heterogeneous multichip architectures. These heterogeneous inter-chip communication requirements can be met in a well-balanced and efficient manner through hybrid wireless interconnection design methodologies like multiple wireless channels, bandwidth division among different transceivers, multi-carrier

modulations, etc. along with combination of different radiation patterns of wireless network.

3.8 Conclusions and Future Work

A hybrid chip-to-chip wireless interconnection is proposed in this work to meet the demanding and diverse communication requirements of heterogeneous multichip architectures. The proposed wireless network combines omnidirectional and directional antennas operating at dual channels for efficient communication between any two chips. The omnidirectional wireless links in CPUs, operating at f_{omni} frequency, provide low latency, low energy communication to any other system component. To provide sustained high bandwidth communication for GPUs, its wireless network is implemented using directional wireless links between GPU and memory operating at f_{dir} frequency. The proposed wireless interconnection speeds up application runtime by $1.15\times$ to $24.42\times$ for system with 4 CPUs and 4 GPUs as compared to baseline network with wired buses and PCIe switches. Using low energy and low energy wireless links, it saves packet energy by 94.4% and packet latency by 58.34% as compared to the baseline interconnection.

The hybrid chip-to-chip wireless interconnection in this chapter and HyWin from chapter 2 overcome the communication challenges of multichip and single chip HSA respectively to provide performance and energy efficient communication. The proposed works can be extended in the future to integrate them together to provide a seamless and uniform wireless interconnection for communication across heterogeneous computing elements on single and multi chip systems. Each processing chip in such a system is comprised of multiple heterogeneous subsystems, with each subsystem made up of same ISA computing elements and its own customized sub-network. Several such sub-networks across multiple chips are then interconnected together using a hybrid wireless network. Realization of such a seamless intra/inter chip HyWin interconnection requires addressing several implementation level challenges viz.,

- Implementation of system wide wireless network across multiple chips leads to interference between intra and inter chip wireless communication and among intra chip communication in different processing chips. Intelligent and efficient designs are needed to minimize or eliminate interference and provide high performance communication among the subsystems across multiple chips.
- Enabling both intra and inter chip wireless transmission at each wireless interface requires modifications to their design, especially for amplifiers and antenna.
- Reliability and security of data transmission over wireless channels pose additional challenges for implementing intra/inter chip wireless networks.

Chapter 4

Wireless Enabled Hybrid Coherence for Many Core Architectures

Cache coherence and associated communication contributes significantly to communication in many core architectures, especially for large applications. Many core systems typically have several cores working in parallel on different application tasks towards achieving high performance. Shared memory systems, the dominant design paradigm for many core processors, enable inter-core communication for parallel execution by sharing the address space among all the cores. Multiple creators, readers and modifiers of data coexist on a single chip, shifting the processors from computation to communication centric designs. Sharing of memory and data among multiple cores requires cache coherence to ensure correctness of multiple data copies. While both software and hardware approaches for coherence have been proposed, the latter have gained prominence due to their superior performance and robustness. Especially, snoopy and directory-based hardware coherence protocols have been widely implemented in a range of multi-core processors. An efficient cache coherence protocol ideally adds minimal overheads, scalable at any system size, while providing accurate sharer tracking and desired performance. In this chapter, we discuss the challenges associated with cache coherence in many core architectures and propose WiSH protocol built over HyWin interconnection for realizing a scalable and efficient cache coherence.

4.1 Cache Coherence Challenges

Traditional ways of implementing existing protocols (snooping and flat directories) limit their scalability, in terms of storage, energy and traffic overheads, with increasing system sizes in many core processors to provide

accurate coherence. Furthermore, coherence traffic adds to the already unscalable communication and interconnect bottlenecks of many core architectures. Hence, intelligent designs to modify existing protocols to improve their scalability and efficiency at any core count are imperative[84].

Snoopy coherence has each core snoop every memory transaction by all cores and generate appropriate coherence messages upon detection of the requested memory block in its cache. It requires high network bandwidth due to large coherence traffic and is unscalable even at moderate number of cores. In addition, ordered networks like buses or crossbars required for in-order delivery of snoopy message, scale poorly for many cores. Network-on-Chip (NoC) infrastructures in many core processors are unordered and do not inherently support snoopy coherence, further limiting its implementation in many core systems. Modifications to snoopy protocol in Uncorq[85] and INSO[86] make it scalable, but still limited to only moderate system sizes.

Directory protocol, on the other hand, maintains coherence by storing sharer information of cached data in a centralized look-up storage. The sharer information is stored either in the lowest level cache tags or directories for higher level caches. It generates significantly less traffic, provides better scalability and is compatible with any interconnection topology, unlike snoopy protocol and has been the preferred choice in most existing state-of-the-art systems. However, the area and energy overheads for accurate sharer tracking using flat directories become excessive as system size increases beyond few tens of cores. Furthermore, the directory access latency and waiting time increase with system size, leading to adverse impact on coherence performance. Optimizations like coarse grain tracking[87], encoded sharer bit vectors[88], alternative directory implementations like Tagless[89] and SPATL[90] and hierarchical approaches[84], etc. have been proposed to make directory protocol scalable to large core counts. Section 4.4 provides a detailed discussion about various existing approaches to improve coherence performance and overheads.

Another major limitation to coherence performance in many core processors is the interconnection network and on-chip communication. High on-chip network latency considerably increases request and response time for remote directories, while making in-order snooping infeasible and adversely impacts coherence performance for either protocol. On the other hand, coherence traffic interferes with data traffic and aggravates the existing communication bottlenecks of many core processors. To this end, cache coherent interconnects are garnering significant interest to improve both coherence and communication performance. These approaches either introduce coherence constraints into NoC design or modify coherence protocols to scale better with NoCs. For example, SCORPIO[91] introduces ordering of snoopy messages within mesh network and ATAC[92] uses optical links with ACKwise protocol to improve coherence performance. In short, an ideal cache coherence adds (i) scalable storage and energy overheads at any system size, (ii) minimal control traffic to NoC and (iii) complements and receives hardware support from interconnection topology.

To achieve the desired targets of cache coherence, we propose **WiSH** (Wireless enabled, Share aware Hybrid) coherence protocol using the hybrid wireless interconnection proposed in this work. WiSH provides efficient, scalable coherence through (i) a hybrid **Snoopy** over **Directory (SpyDir)** protocol (ii) implemented using hybrid Wireless NoC (WNoC) interconnection and (iii) complemented by incorporation of application sharing characteristics into protocol design. The primary aspect of WiSH coherence is a novel hybrid protocol, which is a hierarchical combination of local directory with global snoopy coherence. Traditionally, hybrid coherence schemes have implemented Directory over Snoopy (DoS) protocols, where snoopy protocol operates locally within a small core cluster and global directory protocol maintains coherence among such clusters[93, 94, 95]. In contrast, WiSH implements directory protocol within a subsystem, while inter-subsystem coherence is achieved through snoopy protocol. To the best of our knowledge, WiSH is the first to propose such an implementation. The major challenge for global or in general snoopy protocol in large systems is the ordered network support and high bandwidth requirements. WiSH overcomes this through millimeter (mm-wave) wireless links of HyWin interconnection, which provide bus like ordered, broadcast capable high bandwidth interconnection for global snooping.

The SpyDir protocol of WiSH provides several advantages over conventional hybrid DoS or hierarchical directory protocols viz., (i) significantly low coherence storage and energy overheads (even compared to hierarchical directories) through local directories for small to moderate sized core clusters and elimination of directory at global level, (ii) low network and access latency for directory look-ups within the cluster, (iii) elimination of global request and response messages to remote directories (especially in comparison with DoS protocols) and (iv) low latency and low energy transmission of snoop messages through wireless links for global coherence maintenance. Additionally, SpyDir protocol is easily extensible to multi-die packages and multi-chip systems, where implementation of look-up storage across multiple dies or chips for maintaining coherence is highly difficult. Through optimal design of wireless network (link design, number and placement of transceivers), core cluster size and intra-cluster network, WiSH with hybrid SpyDir protocol provides efficient and scalable coherence for several hundreds of cores.

The coherence performance and overheads are further optimized in WiSH through incorporation of application memory sharing characteristics into SpyDir protocol design. Memory accesses at each core in a parallel application exhibit recurring sharing patterns that allow for optimizations to improve coherence performance or overheads. Proposals like [96, 97] have shown significant improvements to cache coherence by accounting for such sharing characteristics in protocol design. WiSH incorporates similar optimizations to aid SpyDir protocol by identifying blocks as shared within or between clusters and privately cached data at runtime. This is achieved by augmenting SpyDir protocol with a logical, dynamic share aware cache segmentation, that holds coherence

information only for shared accesses across different cores based on sharing status of memory blocks. The protocol state machine is accordingly modified to implement the share aware cache segmentation and coherence maintenance. This further reduces the overheads in WiSH cache coherence. To summarize, the major contributions of WiSH coherence in this work are:

- A novel, scalable hybrid SpyDir protocol with local directory for intra-cluster coherence and global snoop for coherence among the clusters.
- Elimination of directory storage, request and response messages for global coherence across the system to significantly reduce coherence overheads.
- Ordered interconnection support and low latency, low energy transmission of global coherence messages through mm-wave wireless links.
- Share-aware logical segmentation of caches to augment SpyDir protocol and further reduce overheads, especially in the intra-cluster coherence.
- Detailed evaluation of WiSH coherence at different system and cluster configurations and comparison with traditional coherence protocols.

The remainder of the chapter is organized as follows; WiSH coherence with its SpyDir protocol supported by mm-wave wireless network and share aware cache segmentation is discussed in detail in Section 4.2. The experimental setup and evaluation of WiSH with overheads are presented in Section 4.3. Section 4.4 summarizes existing literature for scalable coherence and compares them against WiSH coherence. We conclude our work in Section 4.5.

4.2 WiSH Coherence

This section describes the WiSH coherence for many core architectures; its hybrid SpyDir protocol with on-chip wireless links and share-aware cache segmentation. We also first provide a brief overview of the system and interconnection architecture of the many core processor before delving into the details of WiSH coherence.

4.2.1 Chiplet or Clustered Many Core Architecture with HyWin Interconnection

The chiplet based or clustered many core architecture and its global interconnection network through broadcast capable on-chip wireless links is shown in Fig. 4.1 (we use clusters and chiplets interchangeably within

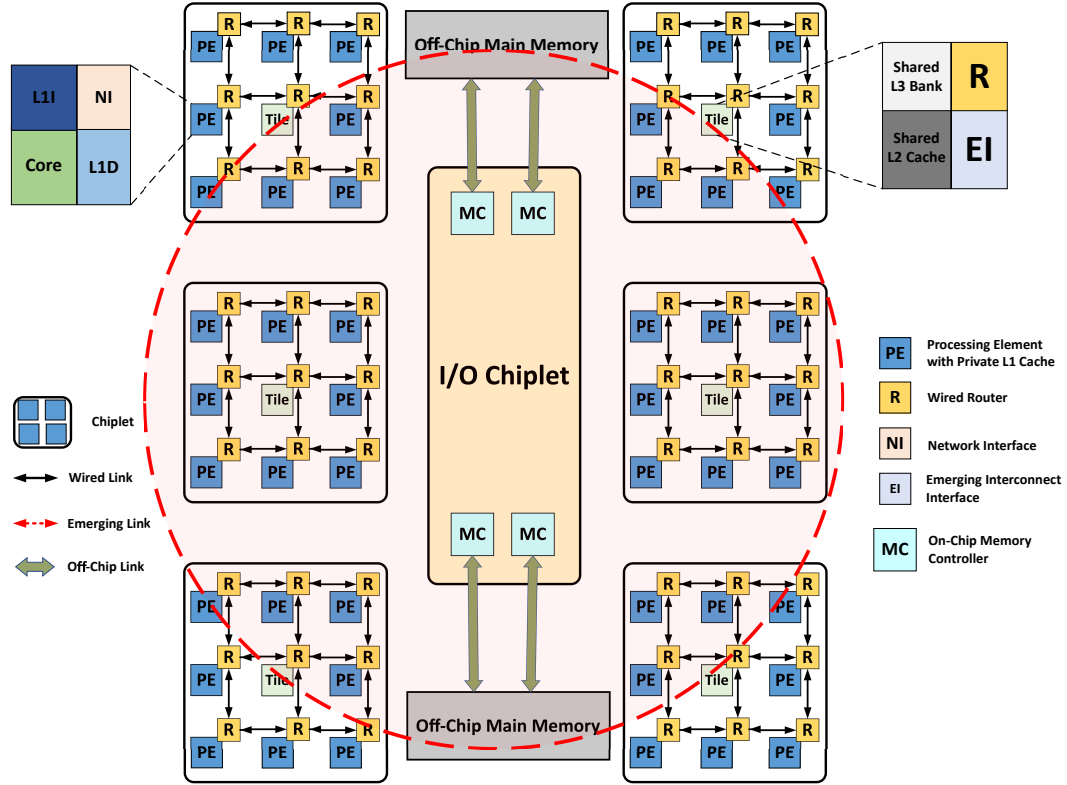


Figure 4.1: Many Core System Architecture with Core Clusters or Chiplets Interconnected by Wireless Links

this context). Many core architectures, with tens to hundreds of cores, can be achieved with low design complexity and benefit from modular and scalable implementation of hierarchical design that integrates smaller clusters of different functionalities and specifications into a large system. Especially, as process technology moves to smaller nodes, chiplet based designs provide better yield as compared to monolithic designs[98]. Through improved yield and cost-effective designs, chiplet architectures allow scaling to large core counts and beyond the reticle limit, not possible with single large dies. For instance, the core complex design of AMD processors[99, 100] and heterogeneous integration of Intel's processors[101, 102] have been shown to achieve high performance, better power optimization and integration following the modular approach. Besides overcoming the physical constraints, chiplet architectures allow integration of dies of different process nodes and companies, optimal for their respective applications[103, 104, 105].

Each cluster or chiplet in the architecture is comprised of neighbouring nodes as show in Fig. 4.1 (illustrated for a 64 core system). Each node in the system consists of a core, its private L1 cache and NoC router for communication with other cores/memory. A single L2 cache is shared by all the cores within a single cluster. A directory is attached with the L2 cache to maintain coherence among the cores that share the L2 cache. While our implementation uses a single physical, shared L2 cache, adoption of WiSH coherence is not impacted by single or multiple banks of shared L2 cache. However, the directory has to be implemented as single physical

structure to help maintain event ordering for global coherence. Finally, a large last level L3 cache is shared among all clusters before accessing the main memory. It is physically banked and each bank is placed within each cluster as shown in Fig. 4.1. All caches along the memory hierarchy are inclusive in nature. The shared L2 cache, its associated directory and shared L3 bank form a single tile within a chiplet or cluster. A snoop controller is associated with each tile in Fig. 4.1 for maintaining coherence among all the core clusters. In chiplet based architectures, a separate chiplet houses all I/O like memory controllers for off-chip components as shown in Fig. 4.1, while I/O is integrated on the same die in monolithic designs.

All nodes within the chiplet or cluster are interconnected using a wired NoC topology, *mesh* in our implementations as shown in Fig. 4.1. On the global level, all clusters communicate with each other through a broadcast capable mm-wave wireless network. The wireless interconnects provide seamless communication between all the chiplets and are highly efficient as compared to electrical links[52]. This high performing, broadcast capable interconnect provides ordered network support for snoop protocol in SpyDir and is akin to a bus in traditional multi-core processors with snoop coherence. A wireless transceiver is implemented with the tile in each chiplet to provide physical interfacing with global interconnect and enable both data and coherence communication with other chiplets and system I/O. It is placed in a way to optimize the access of wireless network to all cluster nodes. Using this modular architecture, WiSH implements the hierarchical, hybrid SpyDir coherence for many core processors (described in detail in section 4.2.2). It is agnostic to all chiplet or cluster design choices like number of chiplets, their sizes and intra-chiplet topology, etc. They can be chosen independently for each chiplet's optimal performance and the global interconnect is designed to provide efficient and low overhead coherence for achieving the best performance.

4.2.2 Hybrid Snoopy over Directory (SpyDir) Protocol

WiSH provides scalable, low overhead coherence for many core architectures through SpyDir protocol, that combines directory and snoopy protocols. Unlike existing hybrid protocols, WiSH is unique in that it implements snoopy over directory protocol and the first to do so to the best of our knowledge. It complements the limitations of either protocol with the other and leverages the advantages of each to achieve this. SpyDir protocol is represented by two hierarchical abstractions as shown in Fig. 4.2.

4.2.2.1 Local Directory Protocol

The inner abstraction circles, each centered around the L2 cache, represent the local directory protocol implemented within the core clusters. It includes the memory hierarchies contained and shared within a cluster

providing 100% coverage using WiSH ($1.3125B/line$) is reduced by $15.71\times$ and $105\times$ as compared to flat ($138B/line$) and hierarchical ($20.625B/line$) directory on the 64 core system shown in Fig. 4.1. The local directory storage for large clusters can further be reduced by adopting alternative directory implementations from previous works like [88, 89].

In addition to storage saving, the local directory protocol also provides latency improvements for coherence performance within the cluster. The latency improvements are achieved in terms of both directory access latency and waiting time. Access to sharer information within the directory requires two protocol hops (different from network hops); cache-to-directory and directory-to-cache for transferring coherence messages. Irrespective of the wired topology of the intra-cluster network, these messages are transferred with low latency from/to nodes within the moderately sized clusters. Furthermore, the small sized directories and relatively smaller number of cores accessing the directory lead to smaller access latency and waiting time to retrieve the desired entries from the directory. Combined, the local directory protocol of SpyDir provides linearly scalable storage and low latency coherence message transfers at any system size by appropriate choice of cluster sizes and number of clusters. These advantages are complemented by the snoopy protocol for global coherence as discussed further.

4.2.2.2 Global Snoopy Protocol

The outer abstraction, shown in Fig. 4.2, represents the global snoopy protocol of SpyDir in WiSH coherence. It is comprised of the second level of memory hierarchies with shared L2 caches of all clusters and the last level L3 cache. To enable this snoop coherence, a snoop controller is associated with the wireless transceiver in each cluster. It snoops all memory transactions between L2 and L3 cache for maintaining coherence among the clusters. The ordered network support for global snoop coherence is provided through the mm-wave wireless network, which is implemented using omnidirectional transmission antennas like zigzag or folded dipole. The wireless network also handles all data communication from/to the clusters in the system.

From the snoop coherence perspective, each cluster acts as single node on the omnidirectional wireless network through its L2 cache as shown by the outer circle in Fig. 4.2 and the wireless channel acts akin to the bus-like interconnect in conventional snoopy coherence. Any miss from the L2 cache or updates to block shared across clusters are broadcasted over the omnidirectional wireless network. The snoop controller at each transceiver intercepts these requests and performs necessary coherence updates to the blocks within L2/L3 caches. This snoop coherence over wireless links is implemented as split-transaction protocol, which allows multiple transactions to be pipelined over the wireless link. A token with round robin arbitration is passed between the transceivers to grant them access to the wireless channel. Any transceiver, after obtaining

the token and has data request, will transmit the request over the wireless link. While the request is being snooped by other clusters, the wireless link is idle and other transceivers can obtain token and transmit their own requests. Different requests and responses are pipelined to make effective utilization of the wireless bandwidth. Any conflicting requests to the same block are disallowed by tracking all active requests in each cluster. Any subsequent updates shared L2 cache or L3 bank to the appropriate blocks in higher level caches within the cluster is performed through the directory protocol. The updates to cache blocks and associated coherence messages are discussed in detail later.

The global snoopy protocol in SpyDir totally eliminates directory at system level leading to considerable storage reduction as compared to conventional directory or even multi-level directory protocols. It eliminates the directory look-ups, request/response messages and associated latencies for performing global coherence by snooping all memory requests from shared L2 caches, leading to improved coherence performance. The low latency of mm-wave wireless network provides an efficient interface for transmitting global snoop messages with minimal delay across the clusters. Additionally, the cores or network within the cluster is transparent to snoop protocol resulting in small snoop traffic, which has been traditionally one of the limitations of the protocol. The only necessary overheads of the global snoopy protocol for controller and wireless transceivers are minimal and tolerable trade-offs for the performance achieved in both coherence and data communication. Through this hybrid SpyDir protocol and interconnect support from wireless links, WiSH coherence leverages the advantages of both directory and snoopy protocols while avoiding their pitfalls to provide scalable coherence in many core processors.

4.2.3 Share Aware Cache Segmentation

WiSH coherence implements different protocols to maintain coherence for blocks shared within or between the clusters and we employ logical segmentation of memory blocks in the caches to identify the shared blocks appropriately. This is achieved through an additional *Sharing Status (SS)* field added to each cache line similar to *Pu* field in directory entry. This logical segmentation is extended to identify privately cached blocks and further reduce coherence overheads in WiSH. Incorporation of memory sharing patterns has long been employed to optimize coherence protocol and several works like [96, 106, 90] have shown significant benefits to coherence through such exploits. The logical segmentation of cache blocks in WiSH allows for a similar implementation in this work as shown in Fig. 4.3.

Analysis of memory accesses by each core in Fig. 4.3a shows that 70% of total accesses at each core, on average, are private. WiSH incorporates this observation into its protocol to further reduce directory overheads

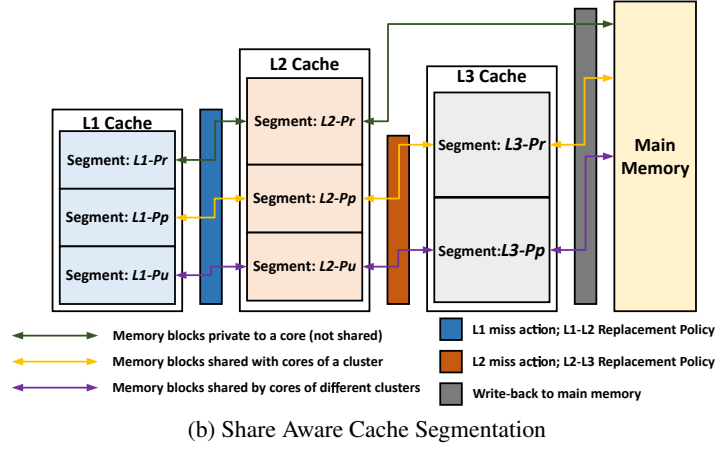
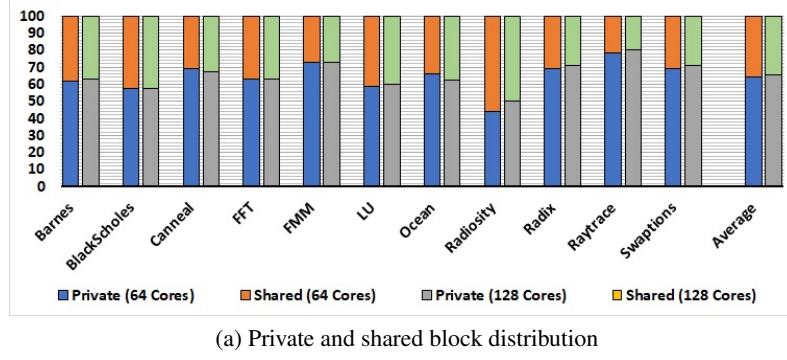


Figure 4.3: Incorporation of Memory Sharing Patterns through Logical Segmentation of Caches in WiSH

through the share aware segmentation of caches as shown in Fig. 4.3b. Based on their sharing status, cached memory blocks are segmented into (i) *Private (Pr)* block cached only in a single private cache (ii) *Partially Public (Pp)* block with more than one sharer and all sharers within a single cluster and (iii) *Public (Pu)* block with multiple sharers across different clusters. This logical segmentation is maintained and updated using the 2-bit *SS* associated with each cache line. As shown in Fig. 4.3b, private caches incorporate all three segments, while L2 cache has only two segments (*Pr* and *Pu*) as all block private or shared within the cluster are private to L2 cache. No segmentation is required at L3 cache as it is inclusive of all cached blocks. Using this segmentation, the local directory protocol eliminates coherence maintenance for all *L1-Pr* blocks to further reduce the storage requirements. Directory entries are only maintained for *Pp* and *Pu* lines. Since, *Pr* blocks make up the majority of accesses at each core, this segmentation leads to considerable reduction of the already small directory storage of SpyDir protocol. Furthermore, only *Pu* blocks require global coherence communication through snoopy protocol, while *Pp* blocks are constrained to intra-cluster transfers and directory updates as described in section 4.2.2.

Table 4.1 presents the coherence states of proposed WiSH protocol. In WiSH coherence, the protocol behaves the same way as the baseline protocol (MOESI in this case) except for one different i.e., coherence

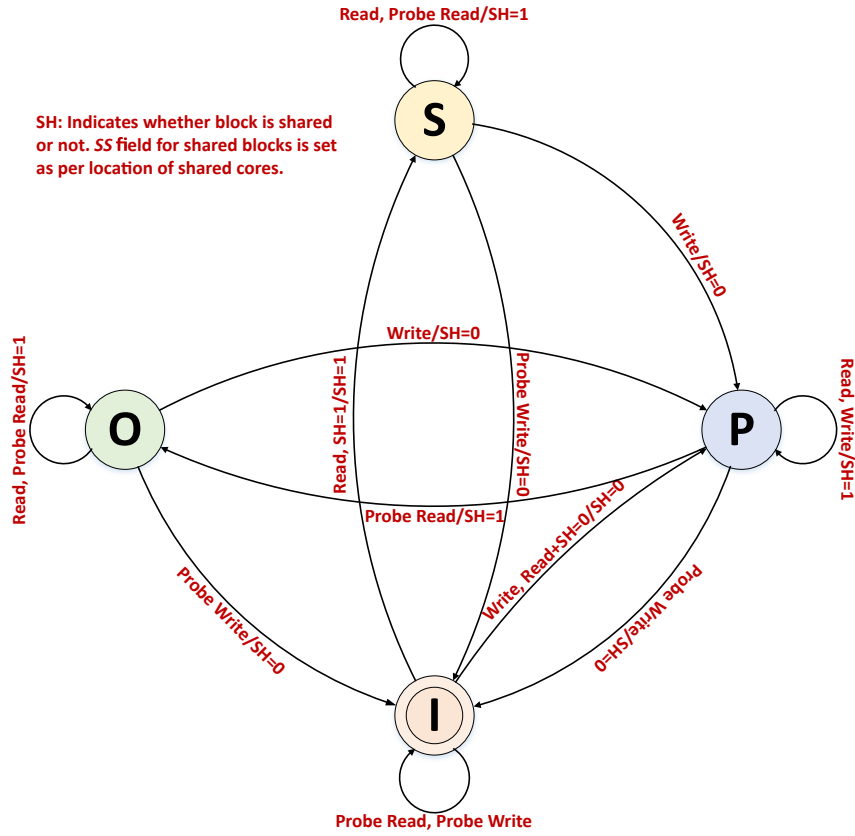


Figure 4.4: State Machine Diagram for Protocol Transitions and Sharing State Field Updates in WiSH Coherence

Table 4.1: SpyDir Protocol States in WiSH

State	L1	L2		L3	Broadcast
		Shared	Others		
Private (P)	I	P	I	I	Cluster
Owned (O)	S/I	O	S/I	S	Global
Shared (S)	O/S/I	O/S	O/S	S	No
Invalid (I)	P/O/S/I	P/O/S/I	P/O/S/I	O/S/I	No

Private is not an actual coherence state and is only a virtual state in WiSH

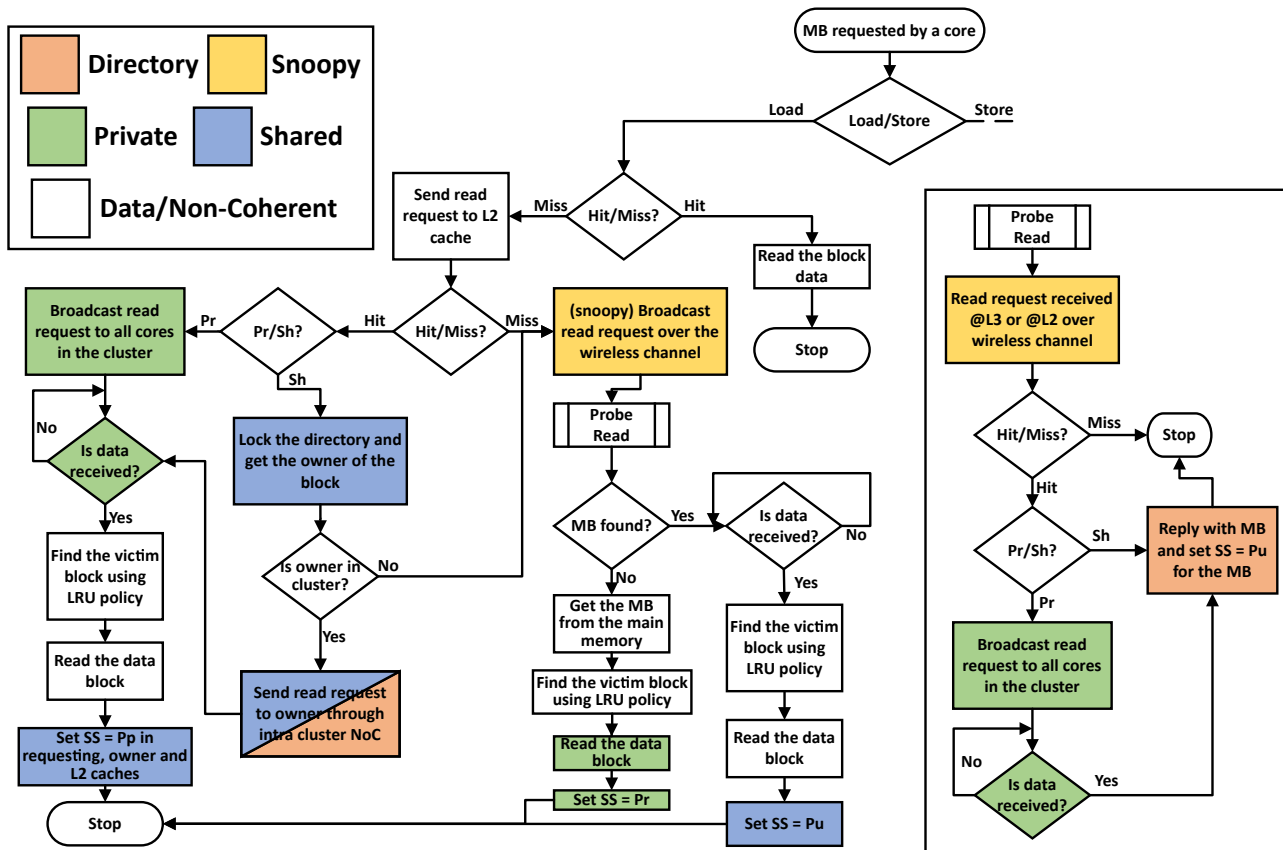
is maintained only for the shared blocks in the caches and no entries are made in the directory for privately cached blocks. The "Private (P)" entry in Table 4.1 is not an actual coherence state and is maintained through *SS* field in WiSH coherence. It is presented here as a virtual state to represent all privately cached blocks for only discussing protocol transitions. The protocol states are comprised of *OSI* states for all shared blocks; "Owned (O)", "Shared (S)" and "Invalid (I)" indicate block owner, block shared and non-cached states like that

in MOESI protocol. The sharers, in case of "S" or "O", are either within same cluster ($SS=Pp$) or different clusters ($SS=Pu$).

Fig. 4.4 shows the protocol state machine for different input requests. All the protocol state transitions for shared/non-cached coherent states, "O", "S" and "I" remain exactly same as those of MOESI protocol. All transitions between private states "M/E" of MOESI protocol and other states require creation or deletion of directory entries as per input request and transition. Hence, the protocol states "M" and "E" are removed in WiSH coherence and no directory entries are created for private blocks. The protocol transitions between different states and privately cached blocks (P) are described in detail further below:

- Any block newly cached into the processor is, by default considered private ($SS=Pr$) as indicated by $I \rightarrow P$ for both read and write requests in Fig. 4.4. No directory entries are created for this block. Any subsequent Read/Write requests to the block from the core where it is cached can be serviced immediately without the need for any protocol updates.
- Any subsequent write (Probe Write) request to a privately cached block from another core/cache invalidates the block in the current private cache and response with data is sent to the requesting cache. The block is also invalidated in the shared L2 cache, if the requesting cache is in another cluster. No directory entry is created for the block in this transition.
- Subsequent read (Probe Read) requests to a privately cached block creates a directory entry for the block, sets the cache containing the block as "Owner" and sets block state to "S" state in all requesting caches. Sharing status is set to Pp or Pu depending on the location of requesting caches.
- Probe Read and Read requests to blocks in "O" and "S" states retain their state, while read miss request sets the state to "S" in requesting cache, while fetching the data from owner cache. These state transitions are same as those in MOESI protocol.
- Write and Probe Write requests to blocks in shared states invalidates the block in all caches except requesting cache (if it is already cached there). The directory entry for the block is deleted and the sharing status is set to private at requesting cache and all shared caches.

Probe Read and Probe Write requests to privately cached blocks from other caches require the request to be broadcasted from the L2 cache to all private L1 caches in the cluster (the cluster where the block is cached). The appropriate cache with the block responds with data and or acknowledgement, while setting the states correctly in its own cache. But, the overheads due to broadcasts are very small as a significant portion of memory accesses are private, resulting in fewer invalidation requests. All protocol state transitions for requests

Figure 4.5: *SS* Field Updates, Directory and Snoopy Coherence Requests for Load Operation in WiSH

within the same cluster are done completely through directory protocol, while those among the clusters require snoopy followed by directory protocol in respective clusters. Elimination of coherence for privately cached blocks has little impact on the protocol implementation and state transitions, however reduces the directory size considerably as the average number of blocks that require directory entries comes down as compared to conventional implementation.

4.2.4 SpyDir Coherence Requests and Status Updates to Cache Lines

The coherence messages generated in WiSH through its SpyDir protocol and sharing status of the cached blocks is shown in Fig. 4.5 and Fig. 4.6 for load and store requests respectively. Upon a read/write hit at L1 cache, the *SS* field of the line is observed to identify its sharing status. If the cache line is private, the desired operation is completed, the state remains unchanged as private and no coherence messages are generated. A read hit request to *Pp* or *Pu* lines also does not require any state changes and no coherence requests are generated. On the other hand, for a store request on hit, an invalid message is sent to all sharer cores. *Pp blocks generate only directory*

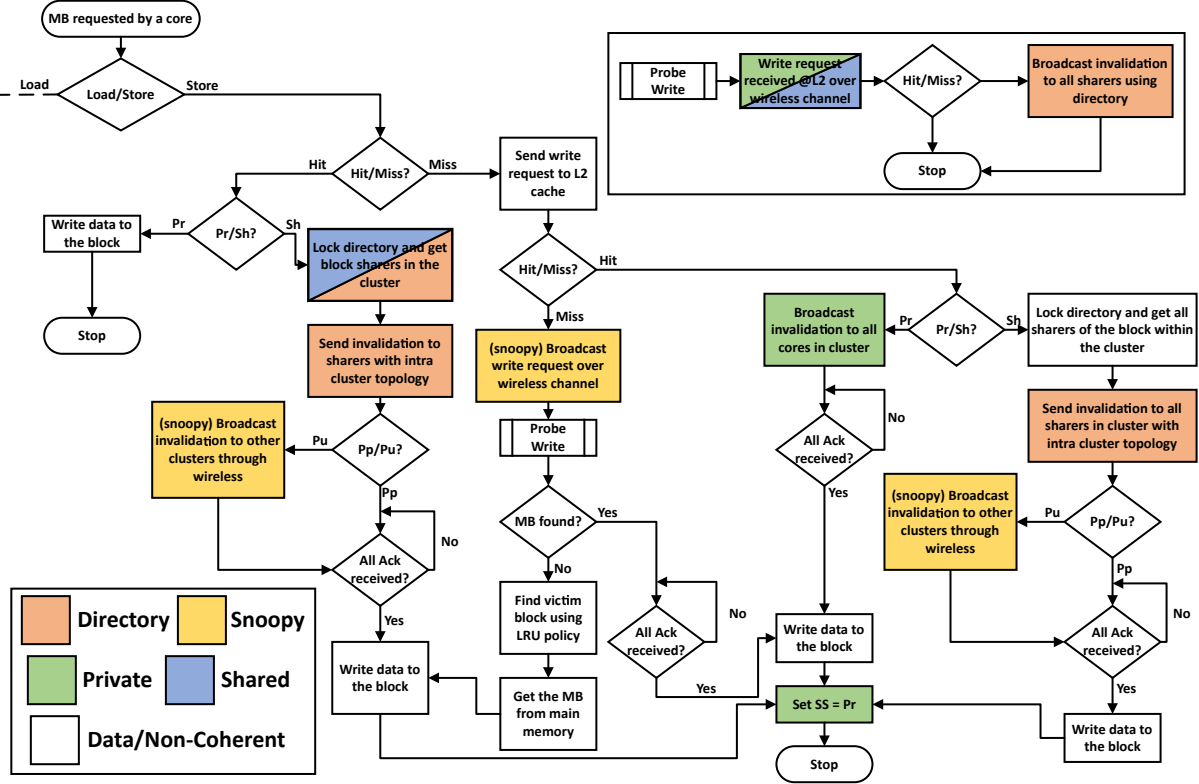


Figure 4.6: SS Field Updates, Directory and Snoopy Coherence Requests for Store Operation in WiSH

requests with invalidates to all sharers within the cluster, while *Pu* blocks additionally transmit the request over wireless network for global snooping and invalidations in other clusters. Upon receiving the acknowledgement from all sharers, the store request is serviced and the block is made private at the requesting cache.

If a request for memory block is miss in L1 cache, the request is forwarded to the associated shared L2 cache. If the request is hit in L2 cache, the directory entry is verified to obtain sharer information and appropriate requests are generated to the sharers within the cluster. A load request generates get request to block owner, while store request generates invalidates to all sharers. Additionally, if the *Pu* bit is set for the entry, the request is transmitted over wireless network for performing global snoop coherence. Upon receiving acknowledgements from all sharers, the directory entry is updated accordingly. For load request, the *Pu* bit for directory entries in both requesting and owner clusters is set, if originally not set and the owner is in different cluster. The sharer vector in directory entry at requesting cluster is updated to include requesting cache as one of the sharers. Similarly, the *SS* for the requesting cache is updated depending on the location of owner cache. For store requests, the block is made private at the requesting cache.

If the request is miss in L2 cache, the request is forwarded to L3 cache and broadcasted over the wireless network to all clusters. If the block is hit in L3 cache, it is obtained through snoop protocol over the wireless

Table 4.2: System Setup for Evaluation of WiSH Coherence

Component	Configuration
CPU Cores	64/128/256 Cores with 8/16 core chiplets, x86-64 Out-of-Order cores, 2.5 GHz
L1 Cache	64 KB, 4-way, 64B line, 1-cycle latency, LRU policy, 2-bit share status/block
L2 Cache	1 MB/core, 8-way, 64B line, 4-cycle latency, LRU policy, 2-bit share status/block, one per cluster
L3 Cache	2 MB/core, 8-way, 64B line, 10-cycle latency, LRU policy, multi-banked

network. A new directory entry is created in the requesting cluster and sharing status is updated at the requesting cache. The directory entry in the owning cluster is updated to set **Pu** bit in case of load request. If the request miss in L3 cache, it is forwarded to main memory and the block is made private across the memory hierarchy.

4.3 Experimental Results

In this section, we discuss the experimental setup for WiSH coherence, including system and network configuration and detailed evaluation of WiSH performance and overheads.

4.3.1 Experimental Setup

The proposed WiSH coherence is implemented with Multi2sim [36] full system simulator using x86-64 cores. Multi2sim is modular, cycle accurate and allows configuration of CPU cores, their specifications and memory hierarchy. It also allows flexibility of specifying interconnection network and routing algorithm for implemented system making it ideal to evaluate the proposed topology. The segmented cache architecture with SpyDir protocol is implemented by modifying the source configurations of memory design in the tool.

4.3.1.1 System Architecture

The details of the implemented system architecture with the cache configuration is shown in Table 4.2. In our evaluations, we have used two system sizes with 64 and 128 cores respectively. Each core has 64KB private L1 cache with 4-way associativity and single cycle access latency. A single L2 cache is shared by all cores within a cluster and a multi-banked L3 cache is shared by all the cores in the system. The sizes and configuration of the shared caches is as shown in Table 4.2. The entire system is divided into multiple clusters with 8/16/32 core cluster configurations. All clusters in our evaluations are of same size, though this is not mandatory for WiSH.

Table 4.3: Details of Network Topology for WiSH

Topology	Wired mesh for intra-cluster network, Token based Wireless for inter-cluster
Routers	5-port, 3-stage (RC, SA, ST), 8-flit depth per port
Wired Link	64-bit width, 1 cycle latency
Wireless Link	60GHz carrier, OFDM modulation, 195.32 Gbps bandwidth, 1 cycle latency
Routing	XY routing for mesh, token based round robin arbitration for wireless links

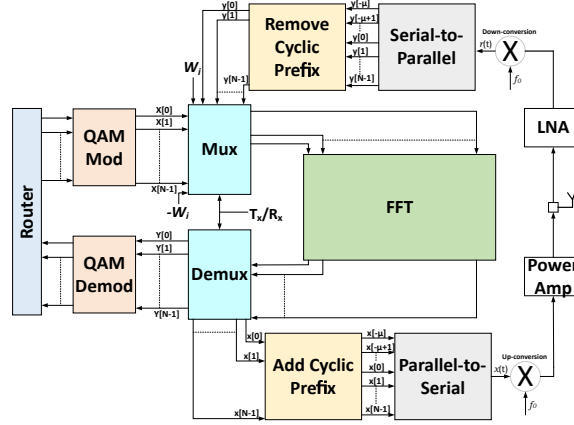


Figure 4.7: OFDM based Transceiver for Inter-Cluster Communication

4.3.1.2 Network Architecture

The configuration of clustered WNoC in WiSH coherence is shown in Table 4.3. As described before, the topology uses wired mesh for intra-cluster communication and wireless links for inter-cluster communication. The wired mesh topology uses 5-port router with three pipeline stages of Route Computation (RC), Switch Allocation (SA) and Switch Traversal (ST). All wired links between the routers are 64 bits wide and have single cycle latency. The wired network operates at the frequency of $2.5GHz$. The intra-chip wired mesh network uses XY routing. The inter-cluster wireless network is implemented using $60GHz$ channel with Orthogonal Frequency Division Multiplexing (OFDM) modulation scheme and on-chip loop antenna[107] with $25GHz$ channel bandwidth. Each cluster is equipped with OFDM based transceiver with token based round robin arbitration for channel access as shown in Fig. 4.7 and explained in detail in 5.11.

4.3.1.3 Benchmark Setup

The evaluation of WiSH coherence is performed by simulating various PARSEC[76] and SPLASH2 [77] benchmarks, shown in Table 4.4, on each system configuration. The application or input sizes for each benchmark are as shown in Table 4.4. The benchmarks are configured to spawn as many context threads as the number

Table 4.4: Benchmarks Characteristics for WiSH Coherence Evaluation

Benchmark	Suite	Application Size
Barnes	SPLASH2	16,384 particles
BlackScholes	PARSEC	16,384 price options
Canneal	PARSEC	2,00,000 elements
FFT	SPLASH2	2,62,144 complex double data points
FMM	SPLASH2	Two clusters, 4096 particles
LU	SPLASH2	512 x 512 matrix, 16 x 16 blocks
Ocean	SPLASH2	130 x 130 grid
Radiosity	SPLASH2	16,384 nodes
Radix	SPLASH2	2,62,144 integers
Raytrace	PARSEC	480x270 Happy buddha object
Swaptions	PARSEC	64 swaptions with 20,000 simulations

of cores in the system in each case. All benchmarks are simulated 10 times and average metrics over all runs are considered for evaluation. WiSH is evaluated in terms of coherence traffic, directory performance, application runtime, network energy and coherence overheads. For understanding the benefits provided by WiSH, we compare these metrics with those of two baseline architectures described further.

4.3.1.4 Directory over Snoopy (DoS) Baseline Architecture

The hybrid **DoS** coherence based architecture uses snoopy protocol for intra-cluster coherence and inter-cluster coherence is maintained through directory protocol. The protocols at both levels are maintained through MOESI coherence states. The ordered network for intra-cluster snoopy coherence is provided through a bus based wired interconnect, which is implemented as a split transaction bus. To minimize the communication bottlenecks of bus based interconnection, we restrict the cluster size to 8 cores in **DoS** based architectures at both system sizes evaluated. The bus in each cluster is also attached to a tile containing single shared L2 cache (shared by cluster cores) and a L3 bank shared by all the clusters. Inter-cluster network, composed of all such tiles, is implemented through wired mesh topology. The global directory provides 100% coverage and is physically banked and each bank is connected to a tile in the inter-cluster network.

4.3.1.5 Hierarchical Directory (Dir2) Baseline Architecture

The two-level hierarchical coherence based architecture, **Dir2** implements MOESI based directory protocol for both intra and inter-cluster coherence similar to that in [84]. The core, cache, cluster and interconnect

configuration for **Dir2** systems follows those of WiSH system described in Table 4.2 and Table 4.3. The intra-cluster directory in **Dir2** follows the same design implementation as that of SpyDir protocol. The inter-cluster directory tracks sharers only among the clusters or shared L2 caches in the system, but not which specific private cache holds the data. Each of its entries consumes one bit per cluster in its sharer vector along with respective coherence state information. The interconnection in **Dir2** system is implemented using the same clustered WNoC topology as that of WiSH.

Besides the baseline architectures discussed above, two other possible hierarchical or hybrid coherence with different interconnect configurations are possible; (i) snoopy-over-snoopy protocol with bus for intra-cluster communication and wireless links for inter-cluster communication and (ii) **DoS** coherence with intra-cluster bus and WNoC for inter-cluster communication. The former snoopy-over-snoopy coherence based system is infeasible because of the limitations imposed by interconnection architecture. The intra-cluster bus is only efficient for small sized clusters, limiting them to a small number of cores (typically upto 8). This results in a large number of clusters as system size increases, leading to significant increase in volume of inter-cluster data and coherence traffic. This makes the inter-cluster interconnection network and global wireless links, a congestion point and performance bottleneck.

The **DoS** coherence based system with WNoC topology for inter-cluster communication improves the overall performance for both coherence and data transfers through use of wireless links for global communication. We have implemented and evaluated This architecture for a subset of benchmarks from Table 4.4 on the 64 core system. While the overall communication and coherence performance is improved over **DoS** coherence with wired interconnect, WiSH still outperforms it in the same vein as that of its performance against **Dir2** system. On average, across the benchmarks evaluated, WiSH reduces coherence traffic by 45.31%, improves application runtime by $1.21\times$ and average packet energy by 50.45% as compared to an architecture with **DoS** coherence and WNoC for inter-cluster communication. In the remainder of the paper, we use the **DoS** based architecture with *bus-mesh* interconnect and **Dir2** systems as baseline for all comparisons.

4.3.2 SpyDir Protocol Performance: Monolithic Architectures

We first evaluate the performance of hybrid SpyDir protocol of WiSH coherence against the baseline architectures in both monolithic and chiplet designs. In monolithic architectures, the cluster size is configured to be 8 cores (8 clusters on 64 core system and 16 clusters on 128 core system) for all three coherence setups. The interconnection topology for **Dir2** is the clustered WNoC topology, same as that of WiSH setup, while **DoS** system uses interconnection setup as described before. The performance is evaluated in terms of coherence

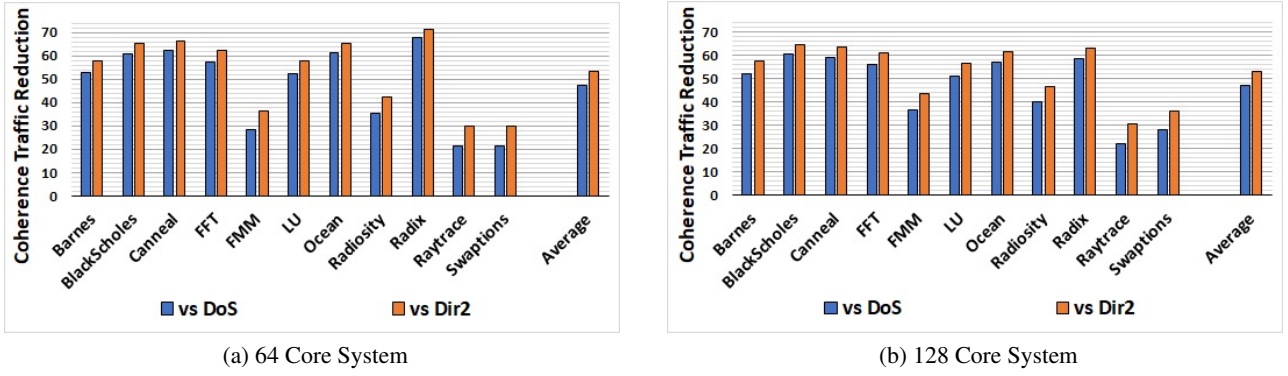


Figure 4.8: Average Coherence Traffic Reduction provided by SpyDir Protocol against **DoS** and **Dir2** Protocols for Different Benchmarks

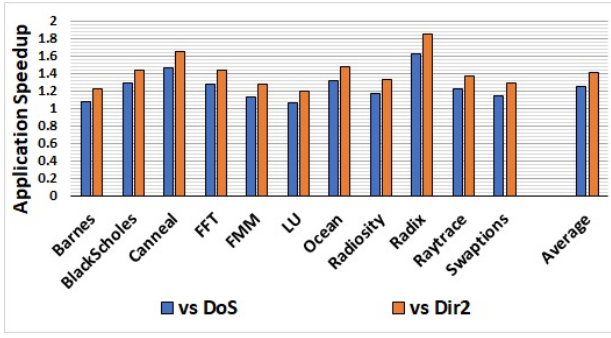
traffic, runtime and network energy.

4.3.2.1 Coherence Traffic

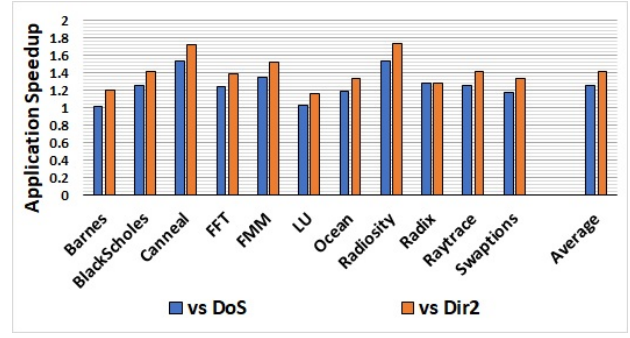
Traffic in coherence mechanism includes directory accesses (if present), evictions due to directory conflicts or spills and write invalidations and adds additional control traffic to the already congested interconnection network in many core systems. SpyDir reduces coherence traffic by eliminating directory and related operations for global communication in the system. Fig. 4.8 shows the average coherence traffic reduction achieved by SpyDir compared to **DoS** and **Dir2** based systems for different benchmarks. The results include all traffic averaged across all caches in the system and are shown for both 64 and 128 core system sizes. On average, Fig. 4.8a shows that SpyDir protocol reduces coherence traffic by 47.44% and 53.20% as compared to **DoS** and **Dir2** protocols respectively on a 64 core system. The reduction in coherence traffic on a 128 core system is 47.54% and 53.35% **DoS** and **Dir2** protocols respectively as shown in Fig. 4.8b. Fig. 4.8 clearly highlights the advantages of hybrid SpyDir protocol in reducing coherence traffic of traditional protocols. While both **DoS** and **SpyDir** use hybrid protocol, unlike the former, SpyDir eliminates directory at global level through snoopy protocol, leading better coherence traffic performance at any system size.

4.3.2.2 Application Runtime

The improved coherence traffic of SpyDir consequently leads to better overall application performance as compared to that of conventional coherence protocol setups. Fig. 4.9 shows speedup in application runtime achieved using SpyDir protocol for different benchmarks on both 64 and 128 core systems. On average across all the benchmarks, SpyDir provides runtime speedup of $1.24\times$ and $1.41\times$ as compared to **DoS** and **Dir2** based sys-



(a) 64 Core System



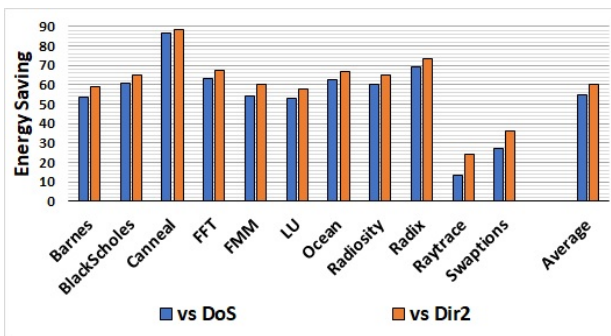
(b) 128 Core System

Figure 4.9: Average Speedup in Application Runtime provided by SpyDir Protocol against **DoS** and **Dir2** Protocols for Different Benchmarks

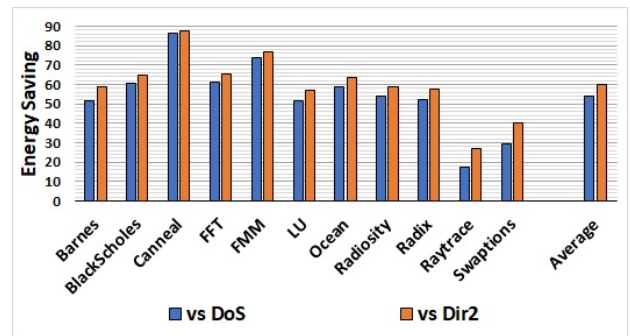
tems respectively as shown in Fig. 4.9a. The speedup is improved to $1.27\times$ and $1.44\times$ against **DoS** and **Dir2** respectively on a 128 core system as seen in Fig. 4.9b. The application runtime receives benefits from low coherence traffic, elimination of global directory accesses and associated access/waiting latency and reduced network latency for accessing remote nodes through wireless links. Similar to coherence traffic, global snoopy protocol of SpyDir provides better advantages as compared to the local snooping in **DoS**. While both **SpyDir** and **Dir2** use wireless links, access, waiting and network latency for remote directory accesses in **Dir2** increases the overall runtime as compared to SpyDir based system. This is also highlighted in network energy savings.

4.3.2.3 Network Energy

Fig. 4.10 shows the energy saving in interconnection network due to reduced traffic and runtime of SpyDir protocol as compared to **DoS** and **Dir2** protocols. The figure shows the average energy for transmitting a packet from any source to destination in the network. On a 64 core system from Fig. 4.10a, SpyDir protocol reduces



(a) 64 Core System



(b) 128 Core System

Figure 4.10: Average Packet Energy Saving provided by SpyDir Protocol against **DoS** and **Dir2** Protocols for Different Benchmarks

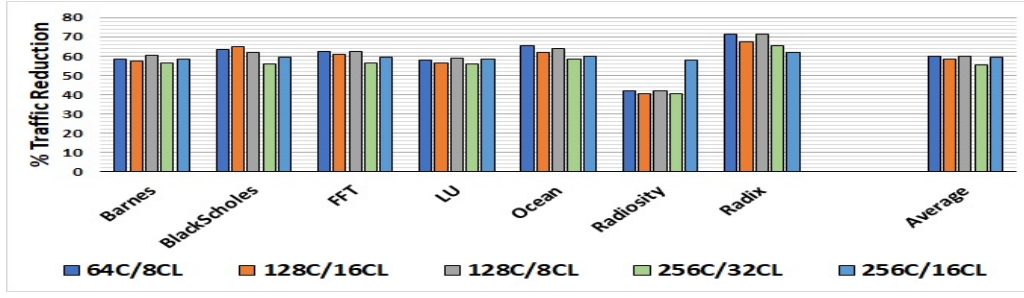


Figure 4.11: Reduction in Coherence Traffic through SpyDir as compared to Two-level Directory Protocol

packet energy by 54.95% and 60.31% across different benchmarks as compared to **DoS** and **Dir2** protocols respectively. Besides the reduction in coherence traffic, the packet energy saving is attributed to both low energy transmission of wireless links and the overall smaller active period of the network from reduced runtime. The packet energy saving on a 128 core system is 54.23% and 59.81% against **DoS** and **Dir2** respectively.

4.3.3 SpyDir Protocol Performance: Chiplet Architectures

In chiplet based architectures, each chiplet houses either 8 cores or 16 cores with the total core counts in the chip varied between 64, 128 and 256 cores. SpyDir protocol performance is evaluated against **Dir2** protocol in terms of coherence traffic, runtime and network energy.

4.3.3.1 Coherence Traffic

Fig. 4.11 shows the average reduction in coherence traffic provided by SpyDir protocol as compared to two-level directory protocol for different benchmarks. The results include coherence traffic across all caches and are shown for different configurations represented as <number of cores>/<number of chiplets>. On average, SpyDir reduces coherence traffic by 60.06%, 60.63% and 59.23% as compared to hierarchical directory protocol on 64, 128 and 256 core systems respectively. The improvements in coherence traffic are better with smaller number of chiplets at all system sizes. Unlike the two level directory protocol which also uses a hierarchical approach, SpyDir eliminates directory at global level leading to better overall coherence performance. This highlights the advantages of global snoop protocol as opposed to global directory protocol.

4.3.3.2 Application Runtime

Fig. 4.12 shows the average speedup in application runtime achieved using SpyDir protocol for different benchmarks on 64, 128 and 256 core architectures with both 8 and 16 core chiplet configurations. On average across

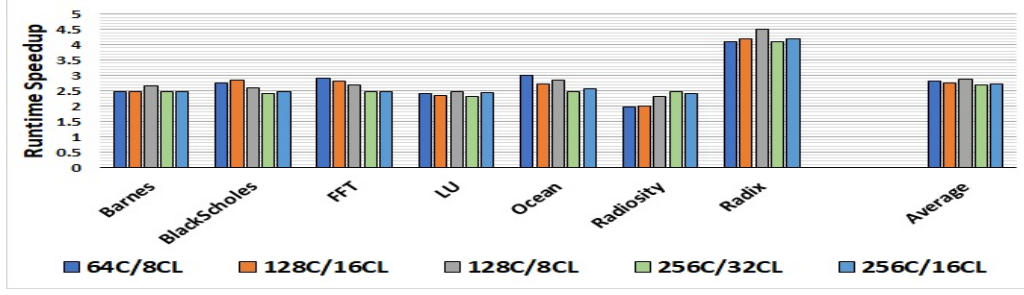


Figure 4.12: Application Runtime Speedup through SpyDir as compared to Two-level Directory Protocol

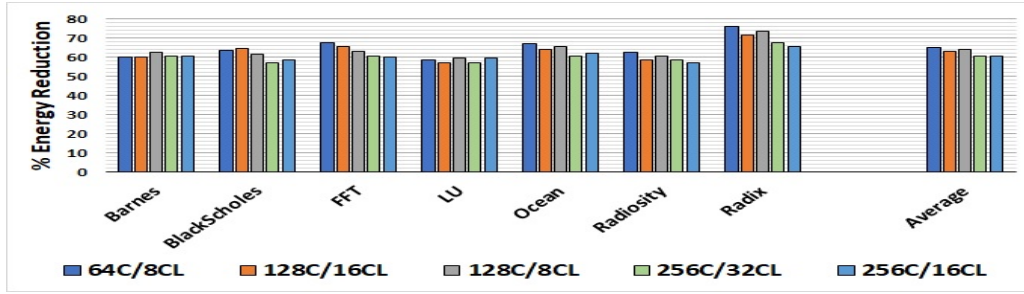


Figure 4.13: Reduction in Network Energy through SpyDir as compared to Two-level Directory Protocol

all the benchmarks, SpyDir speeds up runtime by $2.8\times$, $2.87\times$ and $2.72\times$ as compared to hierarchical directory protocol on 64, 128 and 256 core systems respectively. The application runtime receives benefits from low coherence traffic, elimination of global directory accesses and associated access/waiting latency and reduced network latency for accessing remote chiplets through emerging link. Similar to coherence traffic, global snoopy protocol of SpyDir provides better advantages as compared to hierarchical protocols or local snooping in hybrid protocols.

On a 128 core system, 8 and 16 core chiplet configurations provide runtime speedup of $2.77\times$ and $2.87\times$ respectively. A similar trend of $2.67\times$ and $2.72\times$ speedup with 8 and 16 core chiplets respectively is observed on 256 core system. As the chiplet size increases and number of chiplets decrease, the number of nodes sharing the global emerging interconnect decrease, resulting in lower waiting time and improved runtime performance. A feasible way to reduce congestion on emerging links is to use high bandwidth channels or enable simultaneous transmissions on the global channel.

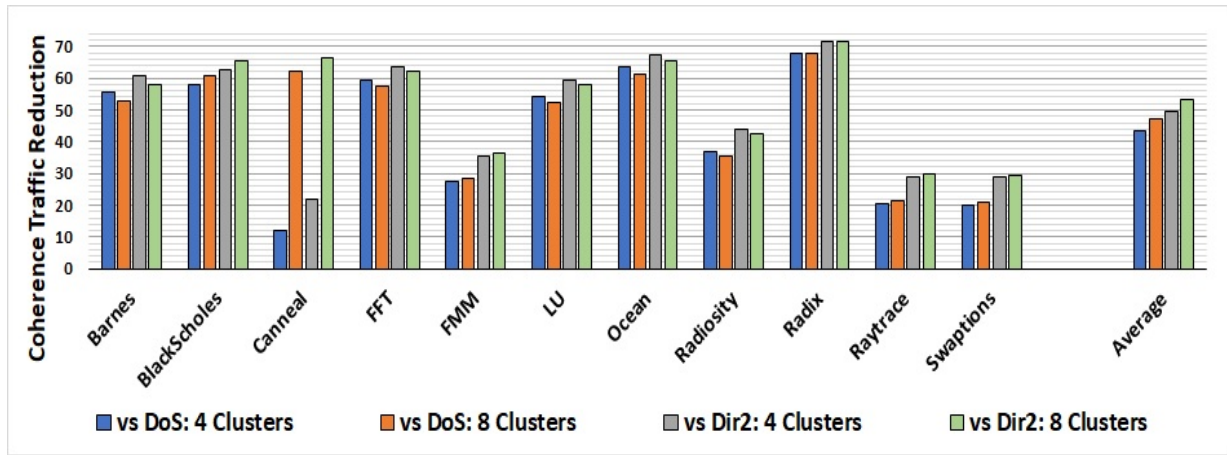
4.3.3.3 Network Energy

Fig. 4.13 shows the energy saving in interconnection network due to reduced traffic and runtime of SpyDir protocol as compared to hierarchical protocol. It shows the average energy for transmitting a packet from any source to any destination in the network within or across chiplets. SpyDir protocol reduces packet energy

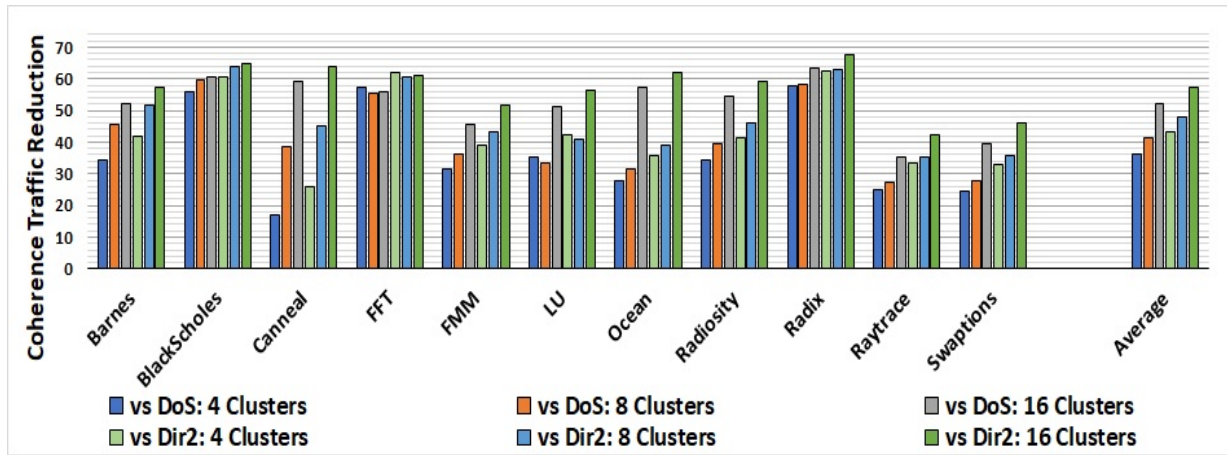
by 65.04%, 63.78% and 60.39% across different benchmarks on 64, 128 and 256 core systems respectively as compared to two level directory protocol. Besides the reduction in coherence traffic, the packet energy saving is attributed to both low energy transmission over emerging links and the overall smaller active period of the network from reduced runtime. Similar to runtime, the congestion and higher waiting time on emerging interconnects with smaller sized chiplets leads to higher average energy consumption as compared to that of larger sized chiplets. Consequently, 8 core and 16 core chiplets reduces network energy by 63.05% and 63.78% respectively on 128 core system and 60.27% and 60.39% respectively on 256 core system.

4.3.4 Evaluation of WiSH Coherence

In this section, we evaluate the efficiency of WiSH coherence with hybrid SpyDir protocol and share-aware cache segmentation. The performance for WiSH is evaluated with different cluster/chiplet configurations; 4,



(a) 64 Core System



(b) 128 Core System

Figure 4.14: Average Reduction in Coherence Traffic provided by WiSH against **DoS** and **Dir2** with Different Cluster Sizes for Different Benchmarks

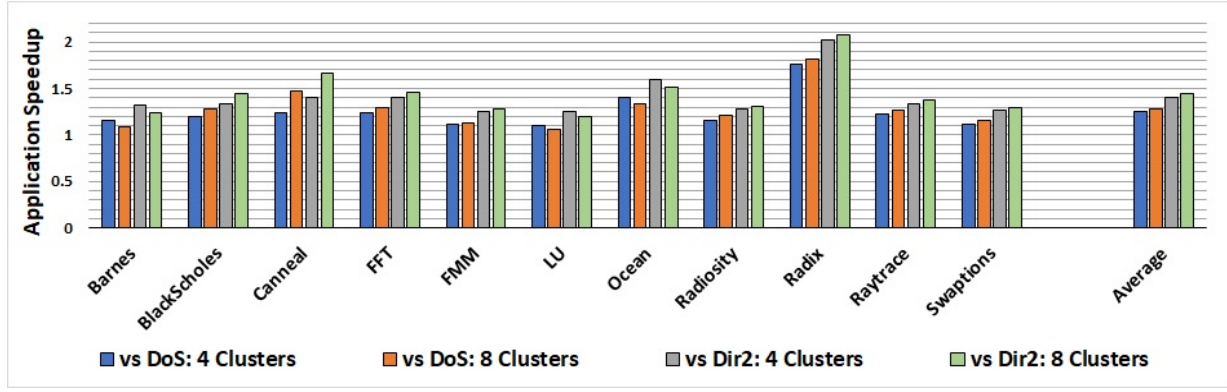
8 and 16 equally sized clusters on 64 and 128 core systems. The baseline system with **Dir2** protocol follows the same cluster and interconnection setup as that of WiSH in each case. On the other hand, **DoS** protocol based system is implemented with only 8 core clusters, similar to that in the evaluation of SpyDir protocol, as bus-based intra-cluster interconnection becomes a bottleneck at larger cluster sizes.

4.3.4.1 Coherence Traffic

Fig. 4.14 shows the coherence traffic reduction achieved using WiSH coherence as compared to **DoS** and **Dir2** coherence protocols for different benchmarks. From Fig. 4.14a, on 64 core system with four clusters (16 cores per cluster), WiSH reduces coherence traffic by 43.24% and 49.46% as compared to **DoS** and **Dir2** protocols. While share-aware cache segmentation eliminates directory accesses for private blocks, it also requires broadcasts within the cluster for these private blocks on probe requests resulting in smaller net improvements. However, these improvements are achieved at smaller directory sizes as highlighted in overheads. The reduction in coherence traffic is further improved to 47.38% and 53.15% against **DoS** and **Dir2** protocols respectively. As the number of clusters increases, coherence communication between remote nodes across the system is transferred to global snoopy protocol leading to improved coherence traffic from reduced directory accesses, conflicts and evictions. A similar trend is observed on the 128 core system with 4, 8 and 16 clusters as shown in Fig. 4.14b. Compared to **DoS** protocol, WiSH reduces traffic by 36.43%, 41.24% and 52.16% with 4, 8 and 16 clusters respectively. The reduction in coherence traffic against **Dir2** protocol is 43.44%, 47.72% and 57.44% for 4, 8 and 16 clusters respectively. The traffic improvements with different cluster configurations clearly highlights the advantages of global snoopy protocol over wireless network as number of clusters increases. However, as number of clusters further increases, the wireless network becomes a congestion point for global communication, leading to adverse impact on the performance. Hence, system design is a balance of cluster size, intra-cluster directory overheads and global traffic over wireless network.

4.3.4.2 Application Runtime

Fig. 4.15 shows speedup in application runtime with WiSH coherence in comparison with **DoS** and **Dir2** protocols with different cluster configurations across different benchmarks. On average across different benchmarks on a 64 core system with four clusters, the application runtime with WiSH coherence is improved by $1.24\times$ and $1.41\times$ as compared to **DoS** and **Dir2** protocols respectively. The runtime speedup is further improved to $1.28\times$ and $1.44\times$ against **DoS** and **Dir2** respectively using 8 clusters on 64 core system as shown in Fig. 4.15a. The application speedup using WiSH coherence is achieved through reduction in coherence traffic, elimination



(a) 64 Core System



(b) 128 Core System

Figure 4.15: Application Runtime Speedup provided by WiSH against **DoS** and **Dir2** with Different Cluster Sizes for Different Benchmarks

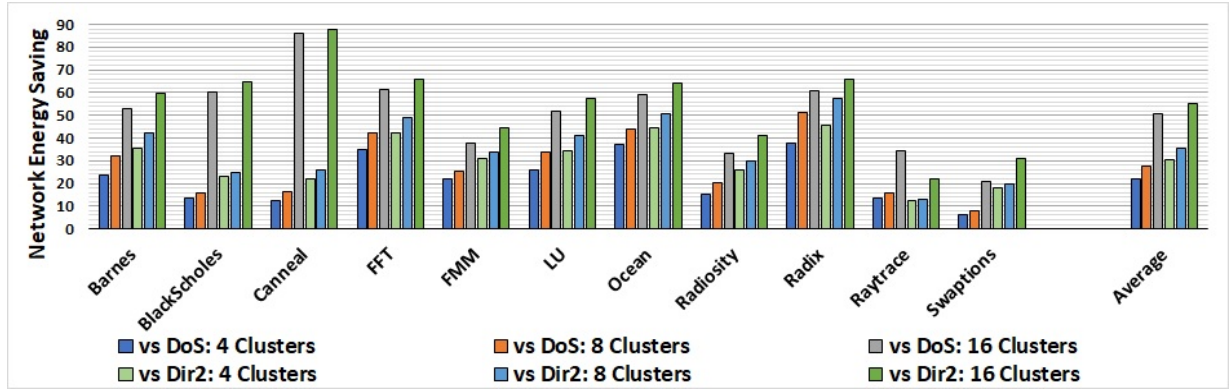
of coherence messages for private blocks in intra-cluster directory protocol and low latency global data and coherence communication over wireless links. On a 128 core system, the application is spedup by $1.15\times$, $1.2\times$ and $1.33\times$ with 4, 8 and 16 clusters respectively using WiSH coherence as compared to **DoS** coherence as seen in Fig. 4.15b. Against **Dir2** protocol, the runtime with WiSH is improved by $1.31\times$, $1.33\times$ and $1.5\times$ with 4, 8 and 16 clusters respectively. Going from 4 clusters to 16 clusters on the 128 core system, the application runtime is further improved by 20% using WiSH coherence in comparison with both baseline protocols.

4.3.4.3 Network Energy Savings

The significant reduction in coherence traffic and application runtime using WiSH protocol and efficient transmission of wireless network reduces network energy considerably as compared to systems with baseline protocols. Fig. 4.16 shows average energy savings in packet transmission across the network using WiSH as compared to both **DoS** and **Dir2** protocols. On a 64 core system with 8 clusters, WiSH saves average packet energy by 55.55% and 60.82% as compared to systems with **DoS** and **Dir2** protocols respectively. The packet



(a) 64 Core System



(b) 128 Core System

Figure 4.16: Packet Energy Saving provided by WiSH against **DoS** and **Dir2** with Different Cluster Sizes for Different Benchmarks

energy saving on 128 core system with 16 clusters is 50.72% and 55% against **DoS** and **Dir2** protocols respectively. Similar to coherence traffic and application runtime, the improvements increase with increase in the number of clusters on both systems as observed from Fig. 4.16a and Fig. 4.16b. As evident from the results, WiSH significantly improves not only coherence performance, but also runtime and network energy as compared to conventional and hybrid protocols using SpyDir protocol supported by global wireless network and share-aware cache segmentation.

4.3.5 Overheads

We evaluate the overheads of WiSH in terms of coherence storage area, network area and power consumption including wireless network components and compare it with those of baseline systems. Table 4.5 shows the overheads of WiSH coherence implementation for different cluster configurations. On a 64 core system with 8 clusters, WiSH coherence requires a total directory storage of 52KB, which is 97% ($\approx 34\times$) smaller as compared to that of full-map directories of **DoS** and **Dir2** based systems. The directory area is further reduced

Table 4.5: Area and Power Overheads of WiSH Coherence

Comp.	NC	Area			E_{dyn} (pJ)			$P_{leakage}$ (mW)		
		WiSH	Dir2	DoS	WiSH	Dir2	DoS	WiSH	Dir2	DoS
64 Core System										
Dir.	4	84 KB	1320 KB	–	52.8	1082.4	–	42.4	3009.6	–
	8	52 KB	1768 KB	1664 KB	31.7	1450	1364	25.44	4031.3	3794
Rtr	–	0.574mm ²		0.215mm ²	0.17/bit			7.537		
$T_X R_X$	–	0.092mm ²		–	0.13/bit		–	25.83		–
128 Core System										
Dir.	4	296 KB	2896 KB	–	190.1	2374.7	–	152.64	6603	–
	8	168 KB	3664 KB	–	105.6	3004.5	–	84.8	8354	–
	16	104 KB	5584 KB	5376 KB	63.4	4579	4408.3	50.88	12731.5	12257.3
Rtr	–	1.148mm ²		0.431mm ²	0.17/bit			7.537		
$T_X R_X$	–	0.092mm ²		–	0.13/bit		–	25.83		–
Comp.: Component; NC: Number of Clusters; Dir.: Directory; Rtr: Router; $T_X R_X$: Transceiver										

to 98.14% ($\approx 53\times$) of that of baseline systems with 128 cores configured into 16 clusters. This shows the considerable storage savings provided by SpyDir protocol and share-aware provisioning of directory in WiSH coherence. A significant portion of the storage reduction is achieved through the global snoopy protocol that completely eliminates directory at global level. The smaller sized directory in WiSH coherence also leads to considerable saving in power consumption of coherence operations leading to highly scalable coherence protocol at any system size. On the other hand, WiSH adds network overheads for implementing wireless network. The OFDM based transceiver adds an area overhead of 0.092mm². The transceivers in all clusters add total overhead of 31.25% over the wired only network of **DoS** system, which is tolerable for the improvements provided by the wireless links for long distance communication.

4.4 Summary of Existing Works and Comparison with WiSH Coherence

Hardware based coherence protocols have been well studied and implemented for processors with fewer number (upto eight) of cores. New approaches and modifications are required to efficiently scale existing protocols many core processors. An ideal coherence protocol has scalable traffic and area overheads and is interconnect friendly at any system size. Fig. 4.17 summarizes the desired coherence requirements and various existing optimizations to achieve those. Most existing modifications for snoopy coherence involve changes to make them suitable for unordered networks like mesh. Techniques like Uncorq[85] and token coherence[108] in-

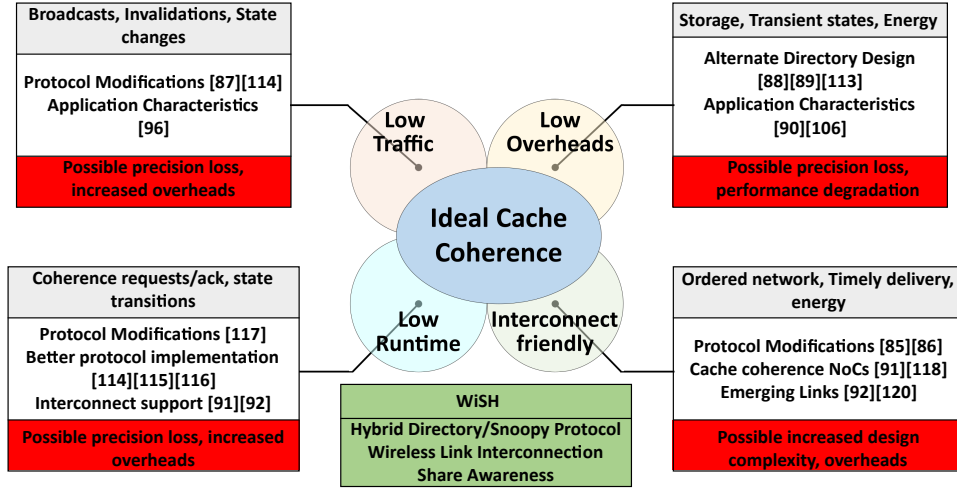


Figure 4.17: Characteristics of Ideal Cache Coherence and Existing Optimizations to Hardware Coherence Schemes

Introduce ordering to snoopy requests for use in unordered NoCs. In INSO[86], interconnect orders all snoop requests as they enter the network. It outperforms traditional directory based coherence by upto 30%. However, the broadcast overhead of snoop requests still adds significant network traffic. Authors in [109] address this by adding filters that reduce redundant requests and achieve 27.3% reduction in network traffic. XPoint cache[110] employs an interleaved cache structure with detailed floorplanning to improve performance of bus based coherence without any modifications at large core counts. It achieves a speedup of $1.6\times$ as compared to conventional bus based design. Even with such improvements, scalability of snoopy protocol is limited only to moderate core counts because of high overheads.

Unlike snoopy, directory coherence stores sharing information in directories and has been the most preferred choice. But, even directory overheads become unscalable at large core counts. Several novel techniques involving alternate directory implementations[88], protocol modifications[111], exploiting application[112] and memory access[87] characteristics have been proposed to overcome these overheads as shown in Fig. 4.17. Tagless directory[89] uses bloom filters for directory to reduce storage overhead without precision loss. It reduces area by 48%, leakage power by 57% and dynamic energy by 44% and incurs only 2.5% additional bandwidth. SCD[88] uses variable size sharer set representation to store coherence data in one tag for fewer sharers and add more tags as number of sharers increases to have $2\times$ smaller storage as compared to hierarchical directories. ICCI[113] leverages the observation that fewer directory entries per-core are required as system size increases to use existing storage structures for coherence and reduce energy by upto 48% as compared to traditional directory protocol. Multi-grain directories[87] track blocks at different granularities to reduce directory size by 41% to 66%. While these works propose alternative directory implementation, optimizations like in [106, 90] exploit application characteristics to reduce coherence overheads. SPACE[106] leverages the

observation that many memory blocks are shared by same set of cores to reduce area overhead and has shown that it requires 25% less area as compared to conventional directory with at most 2% performance degradation. SPATL[90] combines Tagless and SPACE to allow directory compression and further reduces area by 44% as compared to Tagless.

Techniques like [96, 114] make modifications at protocol level to reduce coherence overheads. Authors in [96] exploit the observation that a majority of memory accesses are private and deactivate coherence protocol for these private blocks to reduce status tracking by 57%, leading to runtime improvement of 15%. SPEL++[114] combines weak consistency for race-free code and traditional sequential consistent protocol for racy code to provide optimal performance. Using a static-dynamic classification of code regions to appropriately apply consistency models, it achieves performance improvement of 19% to 38%. An adaptive coherence protocol in [115, 116] combines write-update mechanisms with write-invalidate mechanisms for improved coherence performance and is shown to reduce runtime by 30%. Authors in [117] have proposed a hybrid hardware/software approach through additional directory hierarchy and runtime NUMA-aware scheduling to reduce coherence traffic by upto 77% in NUMA architectures.

Finally, with NoCs becoming prominent in many core processors, cache coherent interconnects and interconnect aware protocol design are gaining significance to improve coherence performance. SCORPIO[91] uses separate fixed-latency, bufferless network for ordering of broadcast coherence messages to allow any time and any order message delivery to reduce runtime by 24.1% as compared to distributed directory protocol. A low power on-chip network in [118] provides guaranteed services for snoopy coherence through a bufferless network for ordered delivery of requests with fixed latency. It reduces latency by 84% and increases throughput by $7.5\times$ as compared to SCORPIO. ATAC[92], a 1K core cache coherent processor uses nano-photonic optics with ACKwise protocol to improve performance by $2.5\times$ as compared to conventional protocols on wired networks. A hybrid NoC [95] for cache coherence combines global directory and local snoopy protocol with dynamic re-configuration of point-to-point NoC links to form local buses for snooping in nearby cores and reduce network traffic by 20% and energy by upto 20%. Similarly, CCNoC[119] designs asymmetric networks for request and response traffic, while hierarchical cluster based WNoC[120] uses wireless interconnects to provide efficient network support for cache coherence.

WiSH coherence proposed in this work takes advantages of both protocol level modifications with hybrid coherence scheme and interconnect support from wireless links to provide low overhead, scalable coherence for many core processors. It presents a novel hybrid protocol, SpyDir that combines local directory protocol with global snoopy protocol to significantly reduce overheads and improve performance. The ordered network support for global snooping is provided by the bus-like mm-wave wireless network. Additionally, wireless

Table 4.6: Summary of WiSH Coherence along with Existing Works

Topology	Performance	Overheads
[91]	12.9% decrease in runtime	10% & 19% lower network area & power
[96]	15% decrease in runtime	57% reduced coherence tracking
[114]	Upto 38% increase in performance	Upto 53% reduction in energy
[118]	9% runtime reduction (compared to [91])	36% lower area, 39% lower power
WiSH	57.44% traffic reduction, $1.5\times$ speedup	55% energy saving, 98% lower storage
The values presented in the table are directly adopted from respective works, which may have different configurations and implementations irrespective of same system size as that of WiSH. The results and comparisons with WiSH coherence indicate the trend of the improvements provided by each work.		

links also provide low latency, low energy transmission of data and coherence messages across the system. The directory overheads are further reduced by share-aware dynamic segmentation of caches to eliminate coherence maintenance for private memory accesses.

4.4.1 Comparison with Existing Works

Table 4.6 shows the summary of WiSH coherence and its comparison with existing works. To ensure fair comparison, we include works that report either application/network performance and energy/storage results for 64 core systems. SCORPIO[91] implements an ordered mesh NoC with bufferless network to support snoopy coherence and reduces runtime by 12.9% on a 64 core system. A low power on-chip network proposed in [118] to provide guaranteed services for snoopy coherence further reduces latency by 84% and runtime by 9% as compared to SCORPIO[91]. [96] deactivates coherence of memory blocks accessed by only one core to reduce coherence tracking by 57% and application runtime by 15% using directory protocol. SPEL++[114] combines weak and traditional consistency protocols based on race-free and racy code segments to improve coherence performance. It uses static-dynamic approach to classify code segments and improves performance by upto 38% and reduces energy by upto 53%. In comparison, WiSH improves runtime by $1.5\times$, coherence traffic by 57.44% and network energy by 55%, while requiring 98% lower storage as compared to conventional protocols. Furthermore, WiSH coherence can easily incorporate existing directory optimizations like [88, 89] to further reduce the overheads as system size increases.

4.5 Conclusions and Future Work

The scalable coherence, **WiSH**, proposed in this work implements novel hybrid protocol mechanism, SpyDir that combines global snoopy protocol with local directory protocol with support from mm-wave wireless network in many core processors. The mm-wave wireless network provides the ordered interconnection for global snooping in SpyDir protocol. To the best of our knowledge, ours is the first to implement such hybrid protocol, unlike existing hybrid protocols with local snooping and global directory protocol. This implementation allows elimination of directory for global coherence to significantly reduce coherence overheads. The wireless network additionally provides efficient data and coherence communication across long distances in the many core processor. SpyDir protocol in WiSH is further augmented by share-aware cache segmentation to eliminate coherence maintenance for private blocks and reduce overheads. Using this wireless enabled, share aware hybrid coherence implementation, WiSH reduces coherence traffic by 57.44%, speeds up the application by $1.5\times$, while required 55% lower network energy and 98% smaller coherence storage on a 128 core system. The evaluation also shows that improvements provided by WiSH are scalable and increase with system size. In the future, **WiSH** protocol can be extended to heterogeneous architectures to maintain cross-architecture coherence in a scalable and efficient manner.

Chapter 5

Wireless Channel Modelling and Link Design

Wireless links and network form the backbone for HyWin intra and inter-chip interconnection and consequently SpyDir coherence protocol. Hence, efficient design of wireless links is of paramount importance for realizing desired communication performance. Towards this end, we first perform a comprehensive analysis of wireless channel characteristics in on-chip environments and using this understanding, we design a robust, fading resilient and high bandwidth wireless link for use in proposed interconnection.

5.1 Wireless Channel Modelling

Wireless Networks-on-Chip (WNoCs) have been widely studied at the topological and circuit level to understand and analyse their communication performance and power[30, 121, 122]. While these works highlight several advantages of WNoC architectures, these analyses are typically not based on physical implementations of on-chip wireless links. Most analyses make the fundamental assumption that signal transmission through on-chip wireless channel resembles that of free space (Air/Vacuum). However, wireless channel characteristics in intra chip environments vary considerably from free space characteristics, leading to mismatch between expected and achievable performance in WNoCs. On-chip wireless channel is plagued by multipath propagations, near field effects, interference from wired links, etc. deviating its characteristics significantly from free space channel. These channel characteristics determine and greatly impact the major properties of signal transmission like loss, delay, etc and has its effects experienced across the entire spectrum of WNoC design from link performance to analog transceiver specifications to wireless node placement and system level design. Hence, there is a fundamental need for detailed investigations into on-chip wireless channel models for both efficient design and accurate evaluation of on-chip wireless communication infrastructure.

There have been endeavours to analyse the characteristics of on-chip wireless channel and its impact on transmission properties. A comprehensive overview of on-chip antennas, their implementation, benefits and challenges is presented in [123]. Kim et. al. in [124] have investigated intra-chip communication at sub-mm frequencies and shown the feasibility of on-chip wireless communication. Kikkawa et al. have studied the effect of silicon substrates on integrated antennas and found that gain can be improved by using high resistivity silicon substrate [125]. Design guidelines for improving antenna efficiency in presence of metal interference structures (in the form of wired links, power grid, etc.) have been proposed in [126, 127]. Different configurations for silicon substrate and ground shield have been explored in [128] for improving transmission characteristics of the integrated antennas. Simultaneously, there have been implementations for design of efficient on-chip antennas and circuitry for wireless links in WNoCs. On-chip integrated antennas for intra chip wireless communication operating in 10's to 100's GHz have been implemented in [129, 130, 131, 131]. A fully integrated phased array antenna and transceiver design for on-chip wireless links operating at 77GHz have been presented in [132, 128].

Besides these, a few works have also analysed the behaviour of signal propagation in intra chip wireless channel. Zhang et al. in [133], by studying propagation mechanisms in intra chip channels, have shown that path loss factor and propagation delay vary significantly from free space transmission. A study in [134] has used a Green's function-based channel model to establish different propagation paths between on-chip wireless transmitter and receiver. However, their research was confined to simple substrate geometries that did not include metal interconnects and other propagation effects. Multiple metal layer stacks in Deep Sub-Micron (DSM) technology and intricate chip geometries have severe implications on on-chip wireless channel resulting in channel fading, dispersion, etc. Proximity of integrated antennas to wired metal interconnects degrades their efficiency significantly, thereby increasing antennas loss and transmission power. Additionally, wireless transmission in intra chip environments is predominantly in near field propagation region, limiting the use of classical analysis based on antenna gain, directivity, etc. which assume far field constraints. None of the existing works thoroughly examine these limitations imposed by DSM technology design parameters and complex intra chip geometries on wireless channel and signal propagation characteristics, that can have significant impact on circuit design, network and system performance.

The objective of our work is to provide a comprehensive and detailed analysis of on-chip wireless channel including the effects of silicon bulk substrate, SiO_2 , metal interconnect stack and their configuration. To achieve a detailed understanding of signal propagation characteristics in on-chip wireless channel with near field and multipath propagation effects, we develop a three-dimension model of the complex chip geometries with metal interference structures. We employ transient domain-based solvers to simulate electromagnetic propagation

Table 5.1: Summary of Existing and Proposed Works for On-Chip Wireless Link Analysis

Ref	Analysis Approach	Intra Chip Setup	Intra Chip Components	Objective	Results
[132, 128]	Simulation/ Measurement	2D	Si Substrate, SiO ₂ dielectric, Ground plane	Antenna/ Transceiver Design	Ground shield above substrate with sufficient distance improves transmission, various substrate and ground shield setups
[124, 125]	Measurement	3D	Bulk + Oxide layer, Backplane	Antenna performance on different substrates	Radiation in on-chip environment is feasible
[126, 127]	Measurement/ Simulation	3D	3 metal layer CMOS chip	Antenna performance analysis and design rules	Avoid metal structures above and parallel to antennas
[133]	Measurement	3D	Si substrate, SiO ₂ dielectric	Frequency and time domain analysis of intra chip channel	Significantly high propagation delay, Surface wave propagation, Metal lines in parallel or normal improve surface transmission
[134]	Simulation	2D	Si substrate, SiO ₂ dielectric	Theoretical analysis of intra chip wave propagation using Green's functions	Surface wave may not be dominant, Guiding layer below substrate improves surface wave
This	Simulation	3D	9 metal layer CMOS chip	Detailed understanding of on-chip wireless channel and design guidelines at all levels	Includes near field effects, Antenna over cavity, high delay spread, path loss and propagation delay, design guidelines

between two antennas in these intra chip environments and study channel propagation physics. Table 5.1 summarizes the existing works and proposed work for intra chip wireless channel analysis in terms of their setup, objectives and findings. While works presented in [124, 125, 126, 127] have included full metal layer stack of CMOS chip, their objective is analyse and improve antenna performance. On the other hand, propagation analysis in [134] provides insights into channel characteristics, but models the chip as 2D multi-layered struc-

ture without metal layer stack. To the best of our knowledge, analysis in [133] includes detailed insights into channel performance through measurements of fabricated antennas on Si substrate and SiO₂ layers and also includes study of impact of metal interference structures on channel characteristics. In contrast, we model the full metal layer stack of CMOS chips, place the antenna in top metal and understand channel propagation within this setup through electromagnetic simulations. Using on-chip wireless channel analysis, we derive channel characteristics like path loss, delay spread, propagation delay and their inter-dependencies, which can then be used to obtain a realistic estimate of wireless link performance in WNoCs. We also propose design changes for implementation of integrated antennas in CMOS chips for improved efficiency. The study provides crucial insights to system designers to optimize design parameters for antennas, transceivers and network topologies on the basis of underlying channel propagation. The major highlights of wireless channel modelling work are:

- Study of intra chip geometries for channel propagation and challenges for estimation of characteristics.
- A detailed multi-layered three dimensional model for intra chip wireless channel, including substrate, metal layer stack, dielectric layers, etc.
- A novel way of implementing on-chip antennas for improving its return loss and bandwidth in presence of metal layer stack.
- Analysis and computation of path loss, delay spread and propagation delay with distance and position of receiving antenna from transmitting antenna in intra chip wireless channel.
- Guidelines for implementing on-chip wireless links and brief analysis of the impact of wireless channel characteristics on network performance and energy.

5.2 Wireless Channel Estimation and Link Performance: Challenges

Data communication using wireless links is dependent on efficient design of transceiver and antenna and signal transmission through wireless channel. Key design choices of transceiver like input transmission power, signal data rate, amplifier sensitivity, along with link latency and loss performance are heavily influenced by wireless channel characteristics. Hence, it is imperative to obtain a comprehensive understanding of on-chip wireless channel for accurate analysis of wireless network performance and optimal tuning of transceiver specifications. Wireless channel, comprised of any and all components between source and receiver antennas, may alter signal path, reduce its power, induce noise, etc. and channel estimation includes analysis of all such effects. On-chip wireless channel, with its complex geometries and materials of different properties, poses many significant challenges for efficient estimation of its characteristics.

5.2.1 Intra Chip Geometries

On-chip wireless channel, unlike most classical wireless environments, is much more complex due to the intricate geometries formed by billions of transistors and their interconnections within the chip. Major components that comprise the wireless channel include bulk substrate generally comprised of silicon, silicon-di-oxide used as dielectric/insulating material and copper metal for wired interconnections. The elaborate shapes, different material properties, dimensions varying from nm to mm make wireless channel very complex, deviating link characteristics significantly from simple free space estimates. It makes wireless channel modeling and accurate estimation of performance significantly challenging.

5.2.2 Near Field Effects

Another fundamental challenge to channel estimation in WNoCs is that wireless propagation between two antennas is in near field/transition region as opposed to far field in more classical setups. Far field constraint for electrically short antennas (largest antenna dimension less than $\lambda/2$) puts minimum transmitting distance between antennas at $r \gg 3\lambda$ [135]. For mm-wave WNoCs, minimum transmission distance between antennas must be much larger than $15mm$. With on-chip wireless propagation distances typically varying from $5mm$ to $25mm$ and multiple antennas in close proximity, intra chip wireless transmission is dominantly in near field region. Unlike far field region, the electric and magnetic fields and their relationship is not well defined in the near field region. This prevents use of traditional loss and delay estimates for on-chip wireless channel estimates. In our analysis, we compute the channel characteristics from the simulated electric and magnetic field at each point in the channel. To be strictly in far field range, antennas operating in sub-THz or THz frequency range are required, which are achievable with Carbon NanoTube (CNT) and require significant changes to CMOS manufacturing process. The analysis of intra chip channel at these frequencies is beyond the scope of this paper and is reserved for our future work.

5.2.3 Multipath Propagation

The complex geometry of intra chip environment, as described, leads to many different propagation paths in on-chip wireless transmission. Besides the line-of-sight component between antennas, transmitted signal can find different paths due to reflections from (a) multiple layers stacked along height of the chip and (b) chip edges in the antenna plane. Along the height, major propagation paths include reflected components from the interfaces of silicon substrate, metal layers, SiO_2 dielectric medium and free space within the chip. The paths

travelled by reflected components vary depending on relative position of the antennas. Different components experience different path loss, propagation delay, and interference from other reflected components, resulting in huge variations in both signal strength and delay at the receiver. This complicates tuning of transceiver parameters for each setup or tuning the design to worst case scenarios leading to underutilized resources. Multipath propagation also makes the channel dispersive and distorts signal shape, requiring reshaping circuits to compensate for delay spread and distortion. This increases design area and power overheads.

5.2.4 Channel Estimation

The intricate chip geometries, near field and multipath effects render traditional free space estimates ineffective for on-chip wireless links and performance evaluation of WNoCs. Without reliable information of wireless channel characteristics, design of antennas, transceivers, and network topologies to achieve desired performance is a challenging task. To overcome these challenges and obtain reliable estimate of wireless link properties, we develop a multi-layered three-dimensional model of intra chip environment to analyse wireless signal propagation between transmitter and receiver antennas. Transient domain, Finite Domain Time Difference (FDTD) technique, often useful for analysing complex electromagnetic problems, is used to study signal propagation. Channel characteristics like path loss, propagation delay, and delay spread are derived from observed results. We, then utilize this analysis to obtain design guidelines and realistic estimates of wireless link and WNoC system performance. We use antennas with different radiation patterns to study different propagation paths and their impact on loss and delay spread. The established channel characteristics are applicable to most existing chip setups and wide range of antenna designs. This allows circuit designers to implement underlying analog and digital logic to work with any design and achieve optimal performance. Once the desired components are chosen, the final characteristics can be obtained by combining channel and circuit properties. The study provides crucial insights into the on-chip wireless propagation characteristics and provides an abstraction of physical link model to be used by circuit designers for tuning the parameters to achieve required efficiency.

5.3 On-Chip Model and Simulation Setup for Channel Estimation

The 3D model, shown in Fig. 5.1 captures all key parameters that influence wireless channel characteristics. The model consists of multiple layers with each layer representing different components of a DSM chip. The layers are stacked along Y-dimension with each layer spanning XZ plane. The devices (transistors, passive components etc.) are fabricated on a silicon substrate. Since transistor device dimensions are of the order of nano-meters and substrate thickness runs into hundreds of micrometers, we model them as a single homoge-

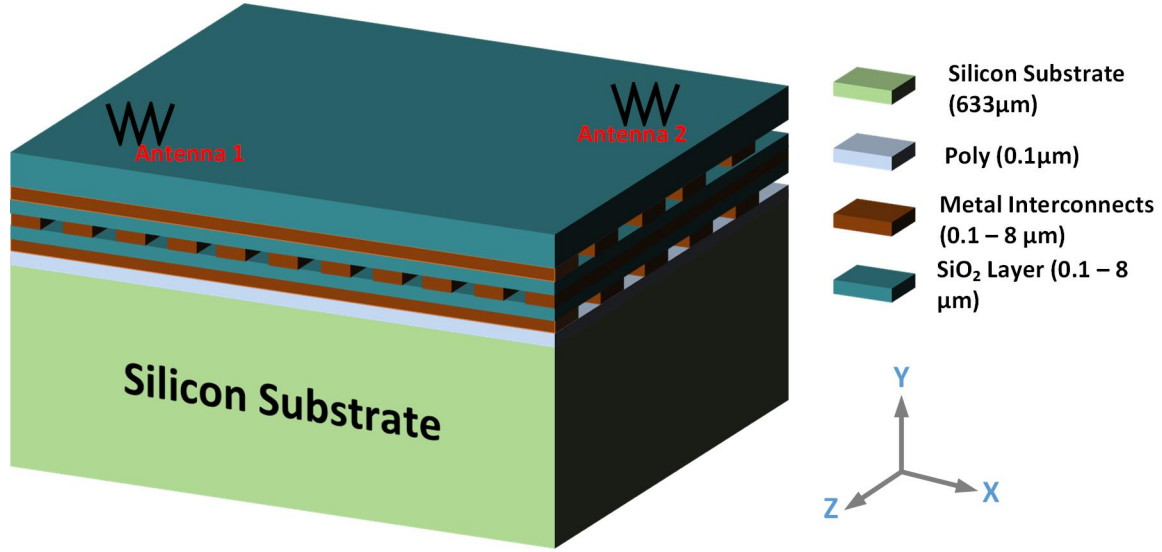


Figure 5.1: 3D on-chip environment with a silicon substrate, copper interconnects and silicon-di-oxide for wireless channel estimation

Table 5.2: Material Properties and Dimensions of the On-Chip Propagation Environment Model

Layer	Dielectric Constant	Conductivity (S/m)	Thickness (μm)
Silicon (Bulk)	11.9	2.5×10^{-4}	633
Metal Interconnects	–	5.8×10^7	0.1 - 8 (increasing from M1 to M9)
SiO ₂ Layer	3.9	10^{-10}	0.1 - 8
Top Dielectric Layer	3.9	10^{-10}	25

neous layer made of silicon, represented as Silicon Substrate in Fig. 5.1. The narrow Poly layer above silicon substrate is representative of poly-silicon used for gate contact in CMOS process. The multiple layers on top of the silicon substrate and poly constitute the different metallic interconnects embedded within SiO₂ material. These interconnects, connecting different devices, span different lengths and are oriented in different directions (X and Z) and are interspersed by SiO₂. We consider the nine-metal layer process (M1 - M9) from 32nm technology and dimensions for each layer are obtained from [136] (Fig. 5.1 is illustrative and does not show all nine metal layers). Metal interconnects in two subsequent layers are separated by SiO₂ material. In each layer, multiple interconnects of different lengths run parallel to each other; with any two interconnects in the same layer separated by spacing determined by the pitch of that layer[136]. The silicon substrate, poly and metal layers interspersed within SiO₂ represent the conventional chip components in any IC.

In on-chip antenna fabrications, typically, antenna is etched in the top metal layer as shown in Fig. 5.1 to reduce the interference from wired interconnects in the IC. Analysis of metal interference structures on antenna performance in [126, 127] shows that metal interference structures above the antenna severely impact the gain and S11 and have to be avoided. Hence, in all our simulations, we place the antenna in the top metal layer

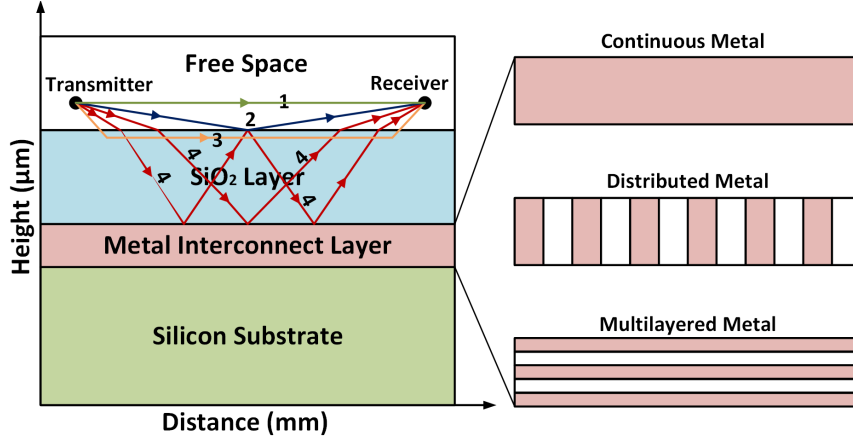


Figure 5.2: Electromagnetic wave propagation in a two-dimensional on-chip environment with a silicon substrate, copper interconnect layer and dielectric layer. The propagation includes following paths: (1) direct free space wave signal from transmitter to receiver; (2) signal reflected off the free space and dielectric layer interface; (3) surface wave propagating through dielectric layer and (4) reflected waves from the metal interconnect layers

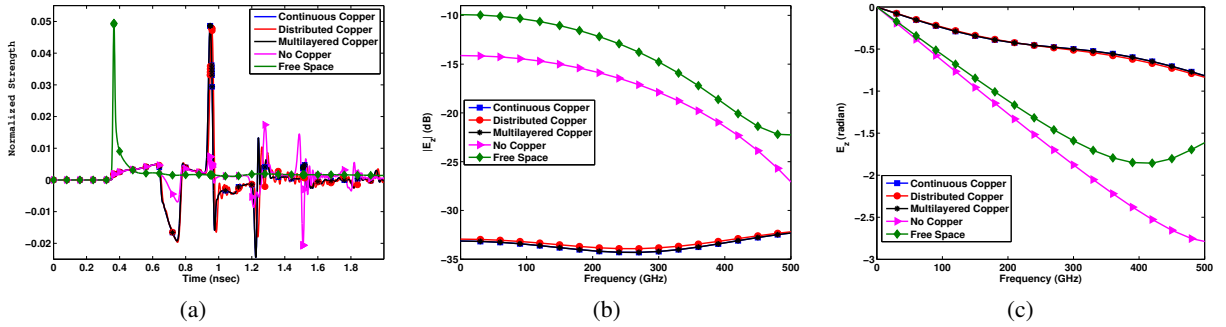


Figure 5.3: The transmitter and receiver are 10mm apart in a two-dimensional on-chip environment with different types of interconnect layer configurations. The figure shows (a) the time domain response (b) frequency domain magnitude response and (c) phase response of the received signal

as shown in Fig. 5.1. We further propose a novel way to implement on-chip antennas for improving their efficiency in presence of wired interconnects, discussed in detail in section 5.5. The antennas and top of model are exposed to free space, representative of vacuum inside the chip. The total 3D propagation model is bounded by Perfectly Matched Layer (PML)[137] bounding region with free space properties to prevent reflections from boundaries of the simulated region. The dimensions and material properties of different layers in 3D chip model are specified in Table 5.2. Here, we ignore the effects of chip packaging because, in most cases, the spacing between the top layer and packaging is significantly higher (of the order of a few millimeters) than the thicknesses of other layers.

5.4 Impact of Chip Layers on Propagation Paths

In this section, we discuss how chip components and design parameters such as the copper interconnect configuration, the thickness of the passive layer and the material properties of the passive layer give rise to different propagation modes in the on-chip channel. For this purpose, we consider the two dimensional lateral view of the 3D chip model as shown in Fig. 5.2. To study the impact of different chip layers, we consider different configurations of the model. In the first case, the simulation space shown in Fig. 5.2 consists of the silicon substrate and the dielectric layers, but without the metal interconnects. In the second case, we introduce the interconnect layer and study its impact on the channel. We compare the time and frequency domain behaviours of signal transmission between two antennas in both the cases, along with that of free space channel. The transmitter and receiver antennas are kept at a distance of $10mm$ from each other. The transmitter is excited using an impulse signal to evaluate the impulse response of the channel in both time and frequency domains and extract different signal components from the received signal.

5.4.1 Time and Frequency Domain Analysis

The time domain and frequency domain response of the received signal at $10mm$ distance from the transmitter for different configurations is shown in Fig. 5.3. In free space, the time domain received signal peaks at $0.036nsec$ as shown in Fig. 5.3a, which corresponds to the distance travelled by the signal in free space between the transmitter and receiver. The frequency domain magnitude and phase responses of this signal are shown in Figures 5.3b and 5.3c. The Gaussian nature of the transmitted signal is reflected in the magnitude response of the received signal, which is weaker than the source. When we consider the case with the silicon substrate and the dielectric layer without the metal interconnect layers, we observe that the received signal shows greater delay ($0.095nsec$) than the free space case. The magnitude response of the received signal in Fig. 5.3b shows approximately $5dB$ additional attenuation in signal strength at mm-wave frequencies.

Next, we include the metal interconnect layers and consider three configurations in the simulation space as shown in Fig. 5.2. The three configurations are considered to evaluate the impact of different layers on the transmission characteristics. Especially taking into the account, the skin effect at different frequencies, if all the metal layers appear as single homogeneous conducting interface or if signal propagates through any of the metal layers. Firstly, as observed from Fig. 5.3a, the delay with all three configurations is considerably greater than that of free space propagation, but lesser than that of when the interconnect layer is absent. Also the strength of the signal (as seen from the time domain response in Fig. 5.3a) is high and comparable with the free space case. Note that this aspect is not reflected in the magnitude response in Fig. 5.3b up to $500GHz$. For mm-

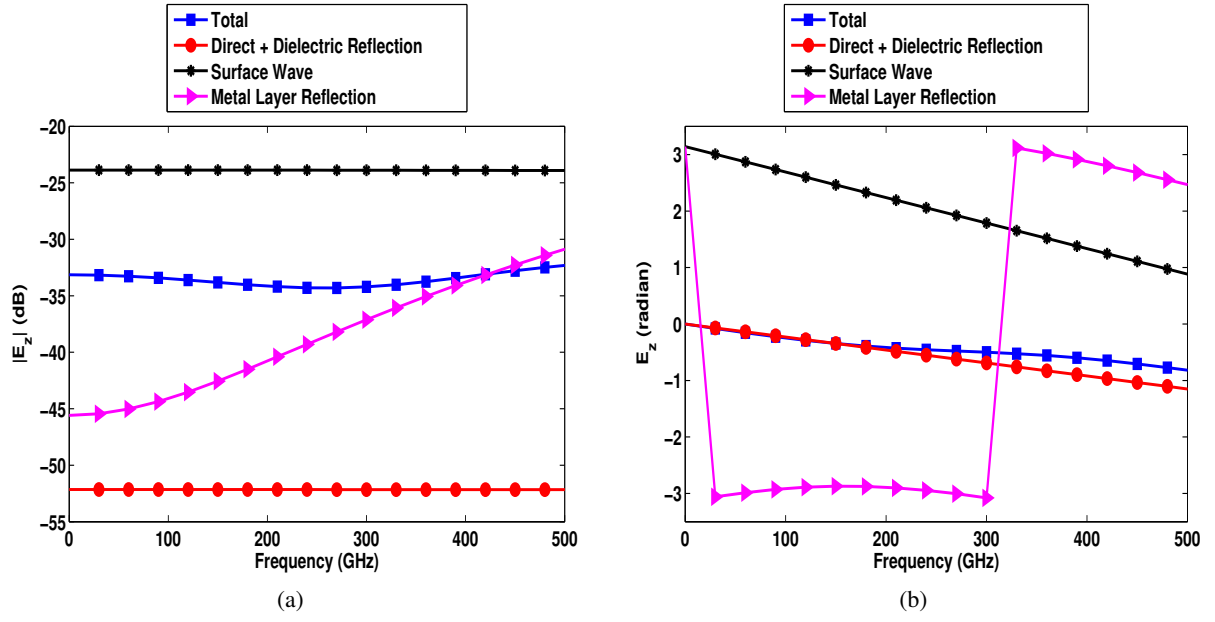


Figure 5.4: (a) Magnitude and (b) Phase response of different received signal components at 10mm distance of from the transmitter in on-chip environment

wave frequencies, the signal strength is considerably lower than that of free space attenuation. At frequencies above mm-wave frequencies (not shown here), though, the magnitude response becomes comparable to the free space results. Most importantly, the three copper configurations do not show considerable variation in the delay, strength or phase response. This implies that for all frequencies in mm-wave and sub-THz range, all the metal interconnect layers appear as single homogeneous conducting layer, which totally reflects the transmitted signal. Hence, no signal transmission exists within the substrate layer of the chip.

5.4.2 Propagation Components

In order to understand the detailed physics behind the observations listed above, we analyse the time and frequency domain behaviours of the received signal to extract different propagation components. The analysis is done only for the case with all layers including metal interconnects are considered. Fig. 5.4 shows the magnitude and phase response of each of these primary signal components. The major propagation components are - the direct wave from the source to the receiver position through free space (ray 1 in Fig. 5.2), the reflected component from the free space - dielectric layer interface (ray 2), the surface wave through the dielectric layer (ray 3) and the reflected component from the metal interconnects (ray 4). Due to the significant thickness of the copper layer, the signal does not penetrate into the silicon substrate layer. When the source and receiver positions are significantly far apart (greater than a few wavelengths), the incident signal on the free space -

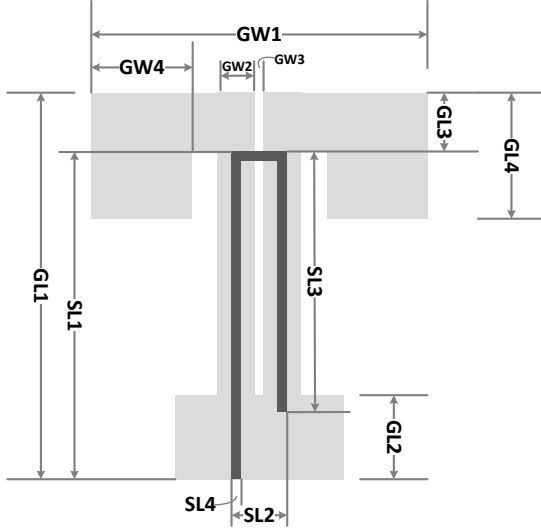
dielectric layer interface is close to 90° and hence the reflection coefficient, Γ , is close to -1 as per (5.1).

$$\Gamma = \frac{\cos \theta_i - \epsilon_2 \cos \theta_t}{\cos \theta_i + \epsilon_2 \cos \theta_t} \quad (5.1)$$

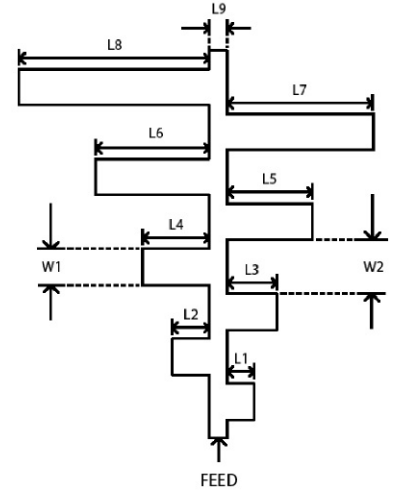
Here, ϵ_2 is the relative permittivity of dielectric layer. θ_i and θ_t are incident and refraction/transmission angles respectively. Note that the reflection coefficient at such large incident angles (θ_i) is mostly independent of the constituent parameters of the dielectric layer. The path lengths of the direct free space wave and the reflected wave off the free space-passive interface are nearly equal at these incident angles and the two components destructively interfere. As a result, the direct free space wave contributes negligibly to the transmission of the signal from the source to the receiver at large distances and the received signal experiences greater delay when compared to free space conditions. One of the possible techniques for preventing the destructive interference of the direct free space wave is to introduce a meta-material based passive layer of zero refractive index $\epsilon_2 = 0$. This will result in a reflection coefficient of 1 at the free space-dielectric layer interface as per (5.1), resulting in constructive interference between the components. The most significant component of the received signal is the surface wave through the passive layer as seen in the magnitude response in Fig. 5.4a. The surface wave travels slower than the free space wave and hence is associated with greater delay in the time domain. The surface wave component consists of multiple modes. The modes in the reflected component from the metal layers are weak at mm-wave frequencies, but become dominant at higher frequencies. From the phase response in Fig. 5.4b, we can infer that the different electromagnetic wave components are not in phase with each other and so the total field is weaker than its components. The analysis presented in this section shows the impact of different chip components on signal propagation through on-chip wireless channel. The different prevailing signal components are same irrespective of antenna structures, their properties and placement and frequencies considered for evaluation. In the subsequent section, we analyse the signal propagation in 3D environment with different antenna structures and their placement. In addition to signal components from different chip layers, this adds signal components from reflections off of various chip edges, further deviating channel characteristics from free space propagation.

5.5 Implementation of On-Chip Antennas

Integrated antennas in on-chip environments experience interference from metal interference structures of wired interconnects used for data communication in MPSoC ICs. Design guidelines in [126, 127] have proposed to avoid metal interference structures above the antenna. Following this, antennas in all our simulation setups



(a) On-Chip Folded Dipole Antenna



(b) Planar Log Periodic Antenna (PLPA)

Figure 5.5: On-Chip Antenna Structures Simulated

are implemented in the top metal layer. To improve the efficiency of antennas in intra chip environments, we propose modifications to existing layer dimensions and a novel way to implement the antenna. Firstly, to reduce the impact of wired links, the thickness of SiO_2 layer between antennas and immediate lower metal layer (referred to as top dielectric layer) is increased. The top dielectric layer has thickness, that does not exceed $10\mu\text{m}$ ($\approx \lambda/500$ for 55-65 GHz frequencies) in today's manufacturing technologies. This significantly reduces radiation resistance of antenna due to its proximity to metal layer and impacts its efficiency. We have varied the top dielectric layer thickness from $8\mu\text{m}$ [136] to $40\mu\text{m}$ and empirically $25\mu\text{m}$ thickness provides the best performance without adding considerable modifications. While this improves the antenna efficiency, interference from metal structures in same layer still poses challenges to efficiency. Furthermore, increasing thickness of top dielectric layer also increases the via thickness and manufacturing complexity.

To improve the efficiency of antenna in practical intra chip environments, we propose a novel implementation to place the antenna over a cavity region formed in between the interconnect layers. This cavity region is completely filled with dielectric material, SiO_2 (generally used material in CMOS manufacturing processes) and no metal structures are allowed within this region. The lateral dimensions of the cavity region in XZ plane are chosen to offset the proximity of any other metal structures by a predefined distance (empirically evaluated to be $100\mu\text{m}$). This reduces the of antenna proximity to all other metal interference structures and improves its gain, S_{11} and bandwidth efficiency. To minimize the impact of cavity region on wired links and their routing, it is formed only in between metal layers M4 and M8 (empirically chosen). In addition, placement of antennas can be chosen to minimize the impact on interconnect routing due to cavity formation. Since, the entire cavity region is comprised of dielectric material, the proposed implementation does not impact vias or contacts

between metal layers.

To show the improvements achieved by using cavity region, we model the placement of antenna and interconnect layers in Fig. 5.1 using three different configurations. In the first configuration, we use the previously analysed setups, where antenna is placed atop silicon substrate and a layer of SiO_2 . This configuration does not contain any metal layers and is referred to as *NoMetalInterconnects* configuration. In second configuration, the antenna is etched in top metal layer; with interconnect layer stack M1-M8 configured as existing in practical chip environments, shown in Fig. 5.1. We refer to this setup as Antenna over Metal Region and is as shown in Fig. 5.1. In the third configuration, we use the proposed implementation; wherever the antenna is to be placed, a cavity region filled with only SiO_2 dielectric medium is created within the interconnect layer stack. The antenna is placed over this cavity region and antenna performance is improved since the proximity to metal interconnects is reduced. We refer to this as Antenna over Cavity Region and is represented in Fig. 5.1.

Furthermore, to demonstrate and understand multipath effects in detail, we use antennas with differing radiation directivities and thereby signal radiated in each direction. For this purpose, two antennas; folded dipole antenna[138] and Planar Log Periodic Antenna (PLPA)[73], shown in Fig. 5.5, are used for our simulations. Folded dipole antenna has a near omnidirectional radiation pattern while PLPA has directional pattern. Directional antennas, that radiate only in specific direction, potentially reduce the number of reflections and amount of signal coupled into different layers, thereby reducing impact of multipath propagation effects. Using the analysis with both antennas, we establish trade-offs between using any of the antenna types and their impact on channel characteristics and ultimately on circuit design and performance. Given the properties of antenna and nature of its radiation, end-to-end signal transmission using on-chip wireless links can be analysed using the characteristics abstracted from channel estimation.

5.5.1 Simulation Methodology

As described in section 5.2, wireless channel estimation in intra chip environments faces several challenges due to complex geometries, propagation in near field region and multipath effects. The sheer number of different on-chip components and several reflections from different layers and chip edges make development of closed form expressions for channel parameters a daunting task. Near field effects further adds challenges to theoretical analysis and efficient estimation of the channel.

To obtain channel characteristics in presence of near field and multipath effects, we simulate the signal propagation between two antennas in the proposed 3D chip model using Finite Differences Time Domain (FDTD) method. We have used the FDTD based transient solver from CST MicroWave Studio (MWS)[139] to simulate

the three-dimensional chip model and wireless signal propagation between antennas. Signal propagation in near field/transition region does not have well-defined electric and magnetic fields and their inter relationship. Hence, we measure the individual field components at each point within the three dimensional multi layered model. The channel characteristics like path loss and delay are then computed from the received field components at different distances from the transmitting antenna. In addition to accounting for near field effects, the field measurement at any point includes all reflected components and constructive or destructive interference among them. This provides for accurate estimation of intra chip wireless channel characteristics.

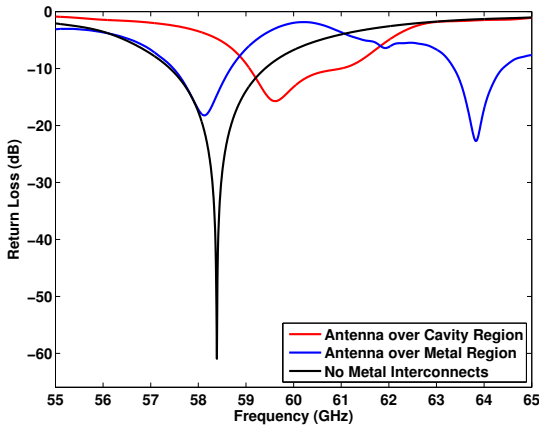
Finally, the propagation analysis is performed at mm-wave frequencies by exciting the transmitting antennas with a Gaussian pulse covering the frequency range of 55-65GHz. The frequency band is chosen because the dimensions of the antennas radiating within this frequency range can be easily etched on top metal layer in existing CMOS processes. Using this setup, the received signal at various distances and for different antenna configurations is analysed to study impact of chip components, near field and multipath propagation effects. The receiver antenna, at any distance, is placed in the end-fire radiation region of the transmitting antenna. The distance measured is from port to port of both the antennas. We derive path loss, propagation delay and dispersion of on-chip wireless channel at desired mm- wave frequencies. The estimated channel properties are then used to evaluate wireless links and network performance. Based on observed effects, we identify efficient ways to circuit implementation and trade-offs between different performance metrics.

5.6 Analysis of Intra Chip Wireless Channel

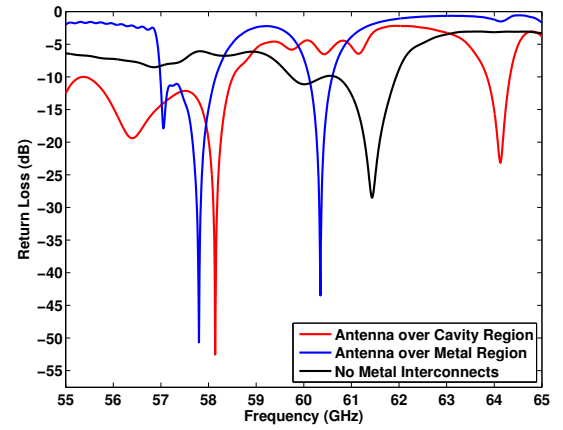
In this section, we discuss the analysis of on-chip wireless channel, its characteristics and how intra chip components effect the properties of channel. We consider distance between antenna ranging up to 20mm which covers most on-chip long range wireless links.

5.6.1 Analysis of Antenna Efficiency

First, we simulate and analyse antenna characteristics in intra chip environments using the three configurations described in previous section. The efficiency of integrated antennas in intra chip environments is degraded due to proximity of antennas to the metal layers. To improve the efficiency of antenna, we propose to place the antenna over a cavity region formed in between the interconnect layer, as shown in Fig. 5.1. The size of cavity in X-Z plane is determined by the antenna dimensions. The length of the cavity in X(Z) dimension is set to be $100\mu m$ plus the longest dimension of the antenna in X(Z) dimension. We study return loss characteristics



(a) On-Chip Folded Dipole Antenna

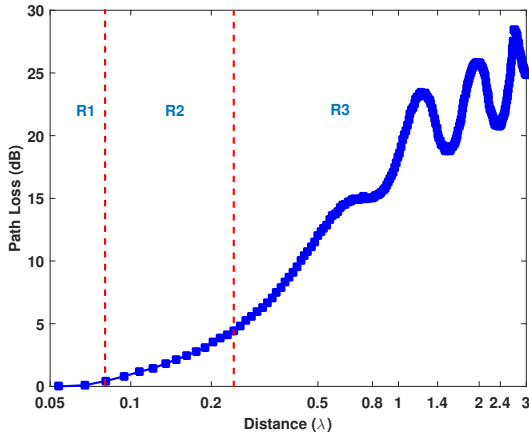


(b) Planar Log Periodic Antenna (PLPA)

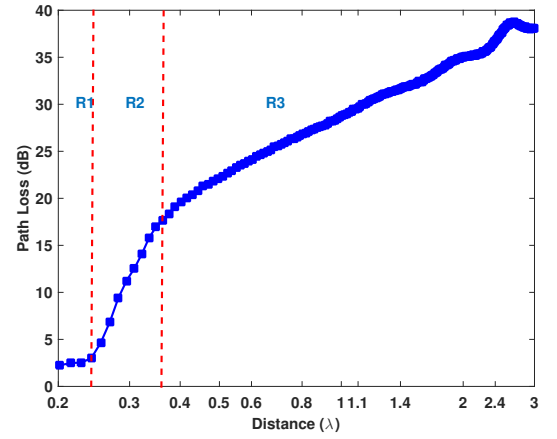
Figure 5.6: Return Loss in Three Dimensional On-Chip Environment for Different Configurations

to analyze impact of metal interconnects on antenna bandwidth. Fig. 5.6 shows the return loss of PLPA and folded dipole antennas for different configurations. The dimensions of antennas in all three configurations are same; only the port impedance of the feed is adjusted in three cases to achieve reasonable return loss values. In No Metal Interconnects configuration, PLPA and folded dipole antennas have bandwidth of 1.5GHz and 2GHz respectively. When these antennas are placed in practical setup of intra chip environments (Antenna over Metal Region configuration), the bandwidth in both cases reduces; 1.16GHz for PLPA antenna and 1GHz for folded dipole antenna. By using dielectric filled cavity where antenna is etched, bandwidth of PLPA is 3.2GHz and folded dipole is 2.4GHz at their respective resonating frequencies. As can be observed from Fig. 5.6, the bandwidth improves by more than 100% while using Antenna over Cavity Region configuration for both antennas. Next, we compare the radiation efficiency of the antennas to observe the impact of metal interconnects.

The simulated efficiency of PLPA and folded dipole antennas, when placed in Antenna over Metal Region configuration, is 61.97% and 60.87% respectively. With Antenna over Cavity Region configuration, the efficiency increases to 94.11% at 58GHz for PLPA and 93.45% at 59.61GHz for folded dipole antennas. We have also verified, if using a top dielectric layer of higher thickness can improve the antenna efficiency in Antenna over Metal Region configuration. Even with increasing the thickness of top dielectric layer from $10\mu m$ to $30\mu m$, the efficiency is increased only by five percentage points. Antenna over Cavity Region provides $1.5\times$ the efficiency of that of existing configuration of metal layers inside the chip for both the antennas. Both radiation efficiency and bandwidth improve significantly when antennas are placed over cavity region, which is completely filled with SiO_2 material as compared to existing setups, irrespective of antenna structure. For the remainder of results, we use Antenna over Cavity Region configuration, unless specified.



(a) On-Chip Folded Dipole Antenna



(b) Planar Log Periodic Antenna (PLPA)

Figure 5.7: Variation of Path Loss with Distance from Transmitting Antenna

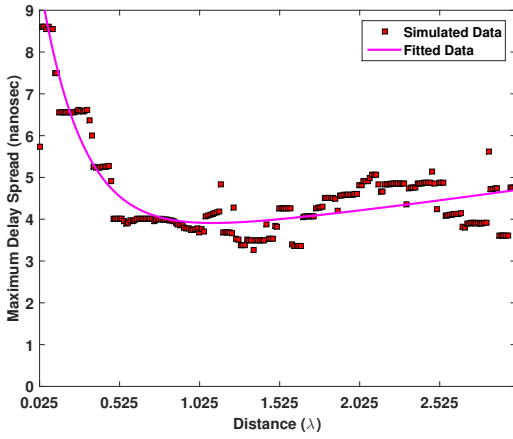
5.6.2 On-Chip Wireless Channel

In this section, we study the properties of on-chip wireless channel and how the chip geometry impacts its characteristics. We derive the path loss, delay spread, propagation delay and their inter-dependencies as functions of transmitter-receiver distance.

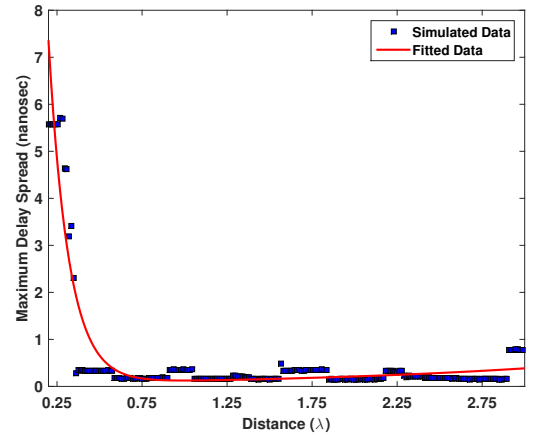
5.6.2.1 Path Loss Estimation

The path loss of wireless channel determines the power requirements for reliably transmitting and receiving data using wireless communication. Intra chip environment with near field and multipath propagation effects degrades the performance of on-chip wireless channel. Fig. 5.7 shows variation of path loss in on-chip wireless channel with distance from transmitting antenna (represented in terms of wavelength) for both PLPA and folded dipole antennas. The field strength decay, unlike in far field region, does not vary as Friis transmission equation with distance from transmitting antenna. To measure path loss in near field region, we analyze the field at receiving antenna for different distances. Due to multipath propagation, signal components that reach the receiver include line-of-sight component, reflection from bottom layers and reflections from chip edges. The propagation components from reflections off of different layers and chip boundaries can interfere constructively/destructively depending upon distance from transmitting antenna.

The variation of field strength with distance can be divided into three distinct regions depending upon distance from antenna as shown in Fig. 5.7. At distances very close to antenna (R1), signal decays at a moderate rate. In R2, with distance upto one-third of wavelength, there is high decay rate in signal strength. Finally, as distance from antenna increases beyond half wavelength (R3), decay rate becomes slower. For PLPA



(a) On-Chip Folded Dipole Antenna



(b) Planar Log Periodic Antenna (PLPA)

Figure 5.8: Variation of Delay Spread with Distance from Transmitting Antenna

antenna in Fig. 5.7b, decay rates for three regions are $19\text{dB}/\lambda$, $57\text{dB}/\lambda$ and $9\text{dB}/\lambda$ respectively. Similarly, for folded dipole from Fig. 5.7a, the values are $16\text{dB}/\lambda$, $27\text{dB}/\lambda$ and $4\text{dB}/\lambda$ respectively. Path loss variation of folded dipole is significantly better than that of PLPA, which is due to higher number of multipath components with former. Omnidirectional antennas like folded dipole radiate in all directions, resulting in many reflected components from all chip edges. These components interfere at the receiver, improving the signal strength. On the other hand, directional antennas result in fewer reflection components as they radiate only in particular direction. Directional antennas, only when placed near chip boundaries experience significant impact from edge reflections. Hence, path loss variation in intra chip wireless channel depends on relative positions of antennas in X-Z plane and antenna radiation beam-width along with channel characteristics. This requires that transmission power at any source needs to be adjusted depending upon receiver position or all transceivers need to be calibrated to worst case path loss. This either increases complexity of design or results in higher power requirements. A possible and efficient way to minimize the impact of this higher power consumption without increasing design complexity is to use low power design techniques like power gating for wireless interconnects[42, 43].

5.6.2.2 Delay Spread Estimation

Another implication of complex chip geometries is time dispersion due to multiple paths and phase distortions caused by it. Delay spread gives the statistical measure of time dispersion nature of channel and is caused by signals from different propagating paths experiencing different propagation delays. As noted before, presence of different dielectric regions (Si substrate, SiO_2), multiple metal layer stack and chip boundaries lead to many propagation paths in on-chip wireless channel. This leads to many signal peaks and high delay spread. Maxi-

mum delay spread is calculated as difference of last and first-time instances of signal paths with strength above a threshold and its variation with transmission distance is shown in Fig. 5.8. We set the threshold as 40dB below transmitted signal power, taking signal loss and noise sources inside the chip into consideration. The directionality and relative positions of antennas impact delay spread of the channel significantly.

As can be seen from Fig. 5.8b, delay spread for PLPA antenna presents an exponentially decreasing trend with increase in distance from the antenna. This can be understood from (a) the number of reflections from chip edges are less for PLPA as it is directional, thereby leading to smaller delay spread; (b) the thickness of different layers along Y-dimension is considerably smaller than transmission distance. Consequently, the path distance of different reflected components from different layers is similar and they experience almost equal delay. This results in reduced delay spread for directional antennas as transmission distance increases. For distances beyond λ , the delay spread is zero as observed from Fig. 5.8b. In case of folded dipole antenna from Fig. 5.8a, delay spread remains high even as distance increases. The high delay spread in folded dipole antenna is due to its omnidirectional nature. Just like with directional antenna, reflections from bottom layers converge. But, signal from omnidirectional antenna radiates in all directions, leading to larger number of reflections from all chip edges. The paths travelled by these multipath components vary significantly and hence they reach

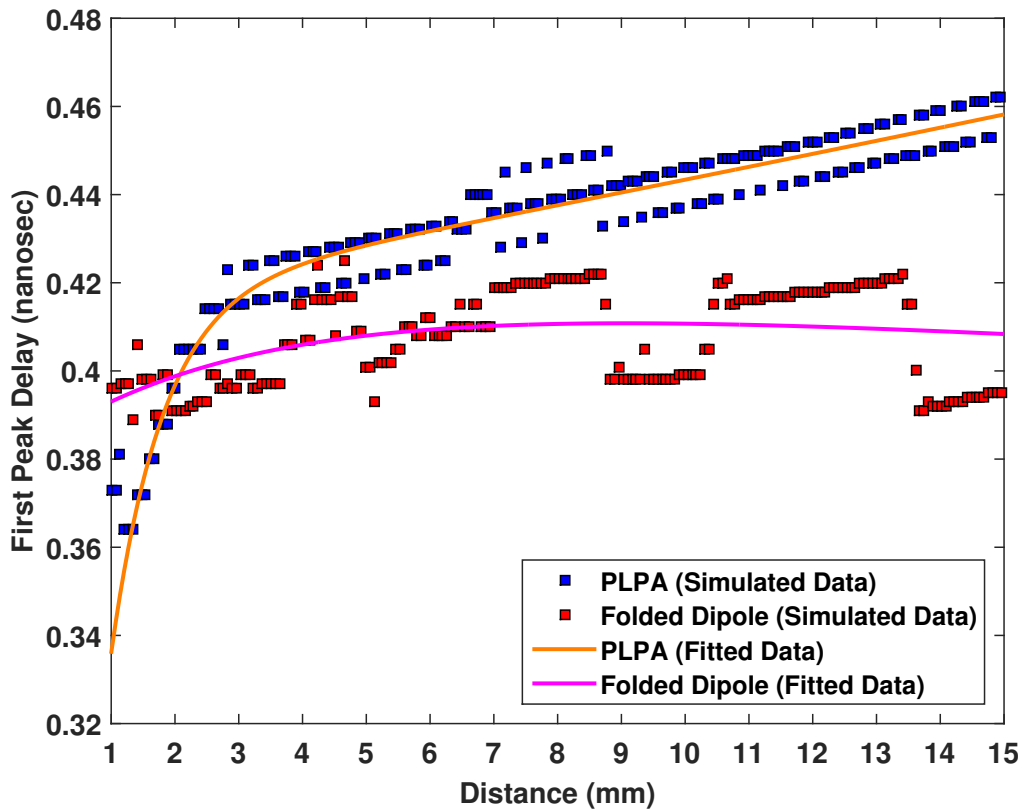


Figure 5.9: Variation of Propagation Delay with Transmission Distance

receiver with different propagation delays. This causes delay spread to be much higher even at longer distances for omnidirectional antenna. Even at distances of 3λ , delay spread is above 3nanosec from Fig. 5.8a.

5.6.2.3 Propagation Delay Estimation

The primary use of wireless transmission for intra chip communications is to reduce communication delay over long distances and overcome the challenges of traditional wired interconnects. We study delay characteristics of on-chip wireless channel with transmission distance. The variation of propagation delay with distance is presented in Fig. 5.9. The delay shows an exponential relation with transmission distance and becomes constant as distance increases. The delay varies significantly from free space delay and varies between 0.36nanosec and 0.46nanosec over distances upto 15mm. In on-chip wireless channel, when source and receiver are far apart, the line-of-sight component interferes destructively with equally strong reflection from the interface of free space and top dielectric layers. As a result, the line-of-sight component contributes negligibly to the received signal. Signal components propagating through and reflections from different layers dominate the received signal resulting in higher delay. Though delay in on-chip wireless channel is higher than free space delay, it is considerably better than that of long on-chip metal wires. At 15mm distance, delay in long buffered and unbuffered wires is 0.9nanosec and 2.4nanosec[140] respectively as compared to 0.45nanosec for wireless.

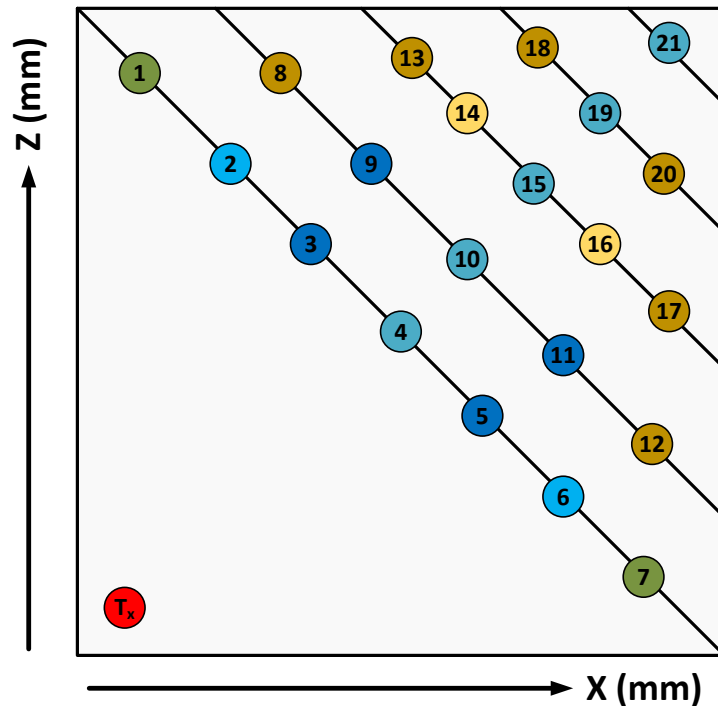


Figure 5.10: Top View of the Transmitting and Receiver Antenna Placements

5.7 Impact of Antenna Placement and Directionality

In this section, we present the analysis of the impact of antenna placement on on-chip wireless channel and different propagation components of the received signal. The three-dimensional chip model from Fig. 5.1 and wireless signal propagation between antennas is simulated using transient solver from CST MicroWave Studio (MWS), which is based on FDTD techniques. The dimensions of the chip are set to $15\text{mm} \times 15\text{mm}$ in the XZ plane and the dimensions of different chip layers are set as shown in Table 5.2.

We have considered both FD and PLPA antennas in our evaluations. The location of the transmitting antenna and the different receiver antenna placements evaluated are shown in the top view of the model in Fig. 5.10. In each evaluation, only one location of the receiver antenna is considered with fixed location of the transmitting antenna. The receiver antenna, at any placement, is oriented to be in the end-fire radiation region of the transmitting antenna. The placements of the receiver antenna are chosen to include different scenarios with antenna away from the edges and closer to one or more edges of the chip. We evaluate the placement and channel characteristics in terms of time domain response, signal strength, delay spread and propagation delay metrics for both the antennas.

5.7.1 Time Domain Analysis

The time domain response of received signal at different placements of the receiver antenna for FD and PLPA antennas is shown in Fig. 5.11 and Fig. 5.12 respectively. As can be observed, several components exist in the received signal due to reflections off of both chip layers and edges. While several unique received signal components exist, both antennas exhibit different characteristics due to their respective directionality.

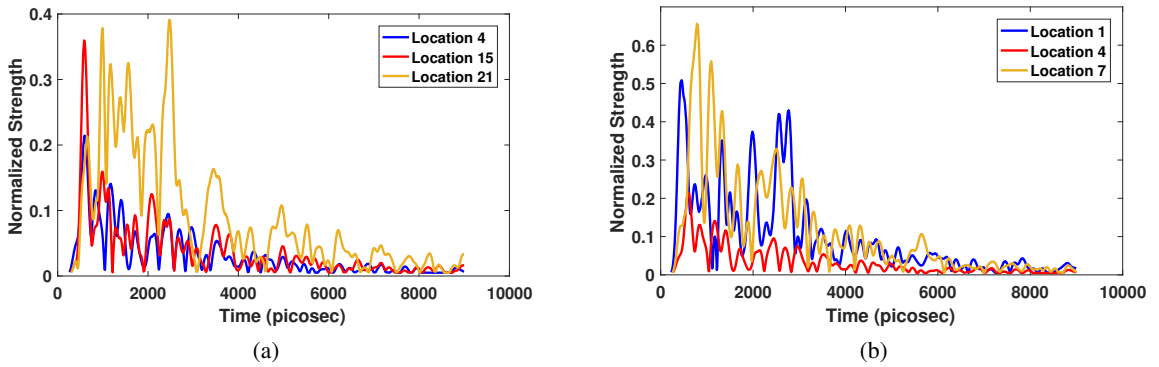


Figure 5.11: Time Domain Representation of the Received Signal for Folded Dipole Antenna at (a) three locations diagonally across the chip from nearest to the farthest distance and (b) three locations orthogonal to the antenna

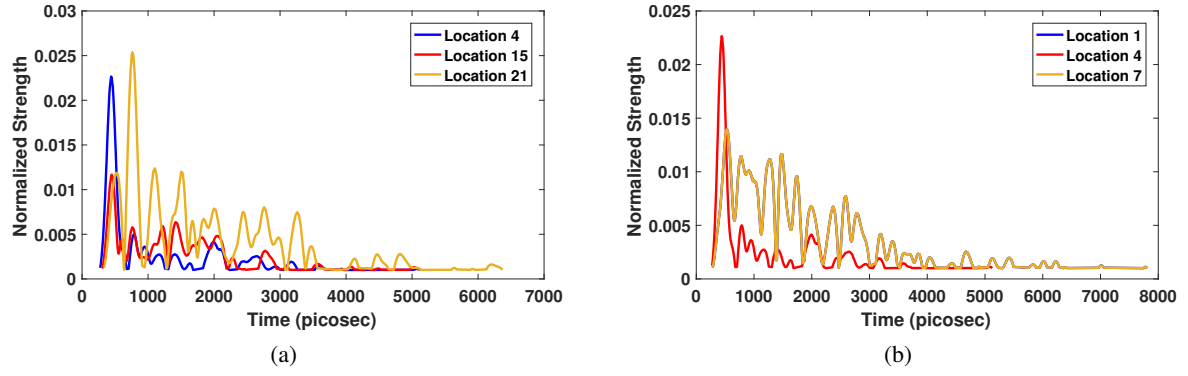


Figure 5.12: Time Domain Representation of the Received Signal for Planar Log Periodic Antenna at (a) three locations diagonally across the chip from nearest to the farthest distance and (b) three locations orthogonal to the antenna

Omnidirectional antennas give rise to many reflections from all edges of the chip unlike directional antennas.

Figures 5.11a and 5.12a shows the response at three locations along the diagonal extending from the transmitter (locations 4, 15, 21) as shown in Fig. 5.10. Irrespective of the antenna directionality, receiver signal at corner placement of the receiver antenna is severely impacted by a high number of signal components. Furthermore, these signal components interfere with each other, giving rise to many components even at different frequencies. Furthermore, the constructive interference between these components leads to signal components comparatively stronger at longer distances. The strength of strongest signal component at location 21 (17mm distance) is approximately twice that of location 4 (8mm transmission distance). A similar variation in time domain response of the received signal is observed in Fig. 5.11b and Fig. 5.12b for locations that are spread along the diagonal orthogonal to transmitting antenna location. Though several signal components exist for antennas placed at chip edges, the number of signal components is lesser as compared to previous case as reflections occur only from one of the chip edges. The number of signal components as the placement of antenna nears chip edges. The major observation from the time domain analysis of received signal with different antenna

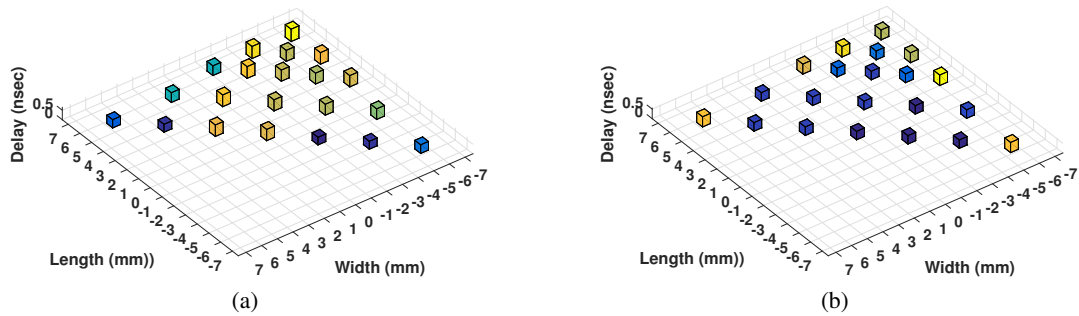


Figure 5.13: Variation of First Peak Delay of the Received Signal with Different Placements for (a) Folded Dipole and (b) Planar Log Periodic Antenna

placement is - while path loss in on-chip wireless channel is significantly high, the antenna placement results in a unique variation of loss with distance due to constructive/destructive interference between signal components. The directionality of the antenna plays a role of the number of prevailing components, but has less impact as antenna placement reaches chip corners.

5.7.2 Propagation Delay

Fig. 5.13 shows the variation of first peak delay with different antenna placements for both FD and PLPA antennas. The propagation delay, with either antennas, at any distance and placement is considerably high as compared to that of free space antenna and agrees with the observation made in Fig. 5.3. The line of sight component between transmitter and receiver provides negligible effect due to destructive interference from reflections off of chip layers.

5.7.3 Delay Spread

A major implication of the highly multipath propagation in wireless channel is time dispersion and phase distortion in the received signal. Delay spread gives the statistical measure of time dispersion nature of channel and is caused by signals from different propagating paths experiencing different propagation delays. Fig. 5.14 shows the variation of maximum delay spread with different antenna placements for both FD and PLPA antennas. The delay spread values are obtained by measuring the time difference between first peak and last peak, whose magnitude is greater than pre-defined threshold. The threshold is set to $40dB$ below transmitted signal peak, taking signal loss and noise sources inside the chip into consideration. Similar to the observations from time domain analysis, delay spread near the chip center is small and it increases significantly as antenna placement

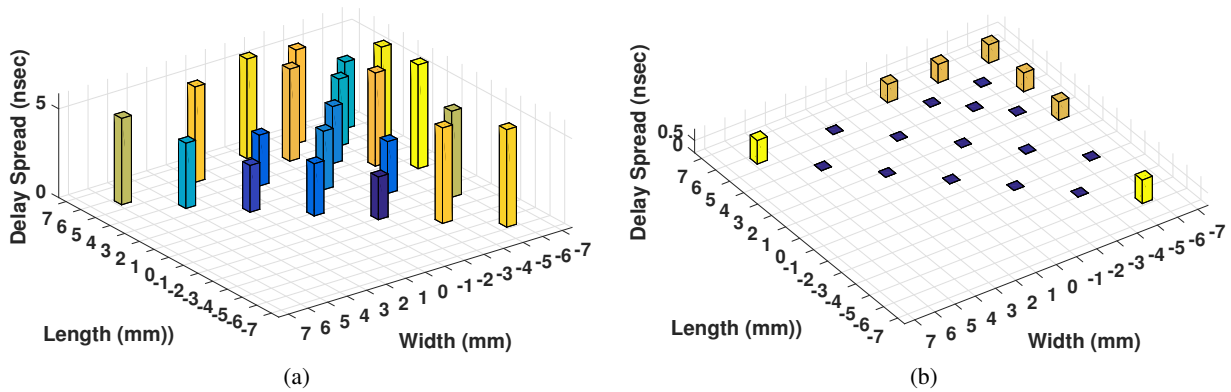


Figure 5.14: Variation of Delay Spread in the Received Signal with Different Placements for (a) Folded Dipole Antenna and (b) Planar Log Periodic Antenna

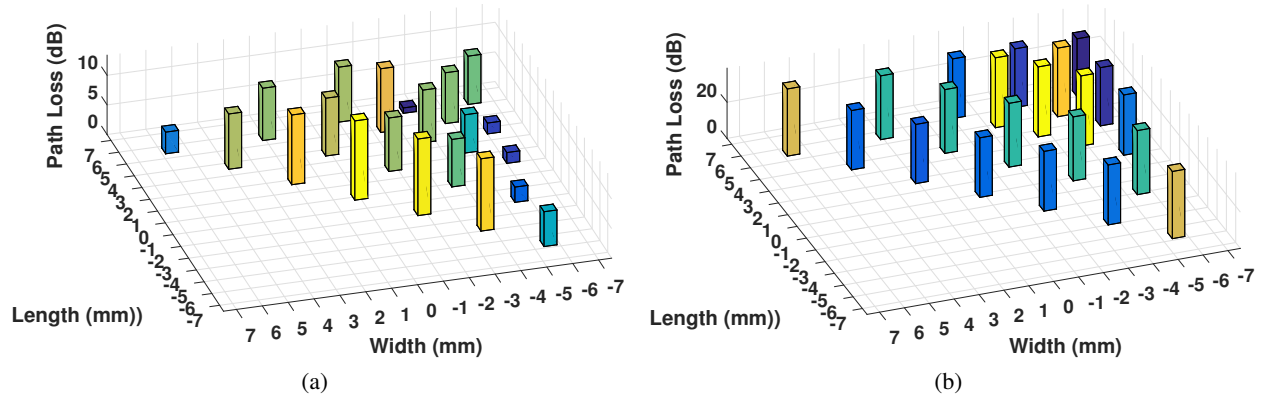


Figure 5.15: Variation of Path Loss with Different Placements for (a) Folded Dipole Antenna and (b) Planar Log Periodic Antenna

nears the chip edges or corners. The directionality of the antenna shows a great impact on the overall variation of the delay spread, along with antenna placement.

As can be seen from Fig. 5.14b, delay spread for PLPA presents an exponentially variation with distance and placement of the antenna. In locations far away from the chip edges, the maximum delay spread approaches zero as (i) no reflections exist due to the directional nature of the antenna and (ii) any prevailing reflections from the edges are significantly smaller in strength and hence provide no contribution to the overall receiver signal. For antenna locations near the chip edges or corners, several reflected components reach the receiver, leading to higher delay spread even with directional nature of the antenna. On the contrary, delay spread for FD is higher at all antenna placements, reaching to the highest values as antenna becomes closer to the chip corners. The high delay spread is attributed to several reflections from all chip edges and interference among them and the maximum delay spread at corners goes as high as $5nsec$. This clearly shows that on-chip wireless channel is highly fading in nature, especially small scale fading, with high dispersion and possibly frequency components not present in transmitted signal.

5.7.4 Path Loss

The path loss in the wireless channel determines the power requirements for reliably transmitting and receiving the signal over wireless link. As observed from magnitude response in Fig. 5.3 and Fig. 5.4, signal propagation in on-chip wireless channel experiences high path loss. Fig. 5.15 shows the path loss at different placements and distances of receiving antenna for both the antennas and validates the same that path loss in on-chip wireless channel is considerably high. Between both the antennas, FD with omnidirectional pattern experiences lower path loss as compared to that of PLPA. This is due to the multiple signal paths, which constructively interfere

with each other leading to stronger signal at the receiver, while using FD antennas. Hence, delay spread and path loss have trade-offs depending on the directionality and placement of the antenna. Omnidirectional antennas have better loss characteristics, while directional antennas have better delay spread and vice-versa.

Another major implication of on-chip wireless channel on path loss is that it does not follow a decreasing trend with distance from the transmitter. As observed from Fig. 5.15, the path loss at the corners of the chip (longer distances from transmitter) is smaller than that at the center of the chip for both the antennas. The high number of multiple paths at the corners of the chip reach the receiver at almost similar path times and interfere constructively leading to lower path loss at the chip corners. This path loss characteristics show an inverse and complex trend as compared to traditional loss variation with distance from the antenna. As observed from delay spread and path loss characteristics, both the antenna directionality and placement play a major role in channel characteristics and have to be considered for topological design of WNoCs.

5.8 On-Chip Wireless Link Design Guidelines

The analysis of wireless channel in on-chip environment shows great deviation from free space channel, previously considered for WNoC evaluations. The intra chip wireless channel characteristics have to be into account for design of transceiver circuit components, wireless network, antennas and their placement and for WNoC performance evaluation. Based on the findings from the intra chip wireless channel analysis, we propose guidelines for WNoC design at all levels from manufacturing, circuit to system design,

1. Any metal lines in the proximity of the antennas, both in the same layer and layers below, must be avoided. An offset of at least $100\mu m$ within the same layer is preferred for all other metal lines.
2. Cavity regions filled with dielectric material in the metal layer stack below antenna structures improve its bandwidth and radiation efficiency. Dielectric materials, besides SiO_2 may further improve performance and require detailed analysis.
3. Multi-carrier modulation schemes like OFDM are suitable for on-chip wireless communication due to resilience against high dispersion and multipath effects prevalent in intra chip wireless channel.
4. A trade-off exists between path loss, channel dispersion, broadcast capability in design of the wireless network.
5. Prudent network design using directional antennas that can establish communication between subsets of wireless nodes may provide robustness against high dispersion.

6. WNoCs, that strictly require broadcast capability between all wireless nodes, can use omnidirectional antennas with bandpass filters on the receiver side to account for dispersion.
7. Irrespective of the choice of antennas, power amplifier specifications must account for transmission power and path loss requirements. While, omnidirectional antennas exhibit better path loss variation, high performance and power overheads may exist for recovering time dispersed signal.
8. The higher power requirements highlight the need for more aggressive power saving techniques, akin to those in [20][21] for wireless transmission and transceivers.
9. High propagation delay and delay dispersion increase the overall channel access time for any single wireless transmission in the network. To ensure reliable transmission in any wireless network design, efficient and dynamic channel resource management techniques are required.
10. The adverse channel effects may require error detection or correction techniques in WNoCs. A detailed analysis of the reliability of on-chip wireless links, including error rate, channel capacity, etc. using end-to-end modelling of wireless communication is needed.

As highlighted by the design guidelines, the effects of wireless channel impact every aspect of system design, increasing the need for robust circuit and network design to ensure efficient and reliable WNoCs. Especially, system level studies must ensure incorporation of all these analyses and guidelines for both design and evaluation of WNoCs. In the next section, we provide a brief analysis of wireless link and network performance with free space channel assumption and intra chip channel characteristics.

5.9 Impact on Wireless Interconnect Performance

In this section, we briefly study the performance of wireless interconnects with free space and intra chip wireless channel. Data transmission using wireless links typically involves data conversion between digital and analog, modulation/demodulation, amplification and signal transmission through wireless channel. Different modules associated with wireless communication are shown in Fig. 5.16. Serializer/Deserializer (SERDES) buffers at the interface of WI converts data between n-bit parallel form and serial data stream. The serial data stream at transmitter is modulated and amplified before feeding into the antenna for transmission. We have used On-Off Keying (OOK) modulation scheme adopted from [23] for this evaluation. The received signal at the receiver is amplified and demodulated before converting back to parallel data. We find link latency and energy consumption using on-chip wireless channel and compare the values with that of wireless link using free space channel and wired links.

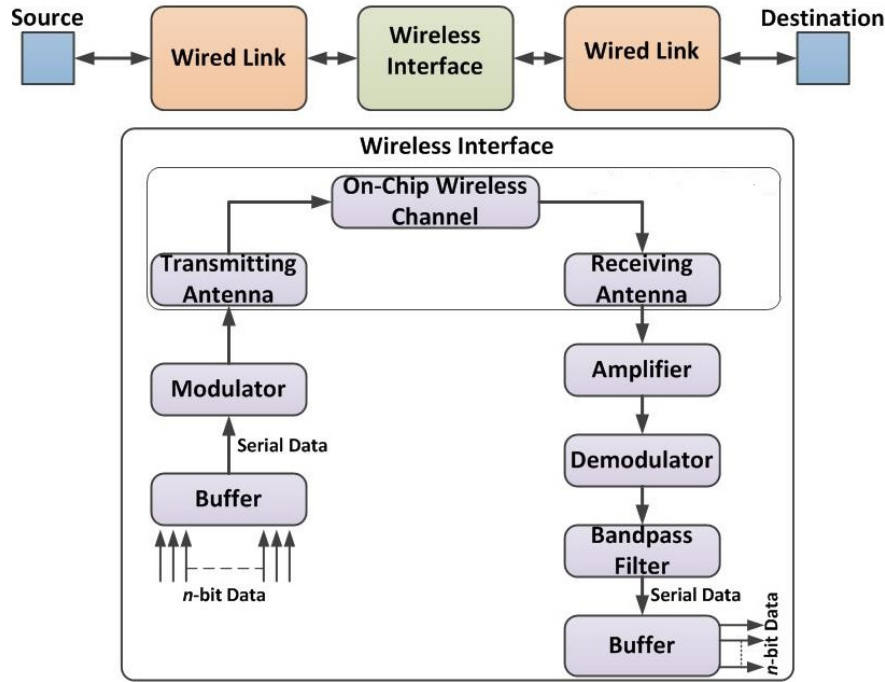


Figure 5.16: A Typical Communication Path using Wireless Interfaces with all Components in NoC

Table 5.3: Comparison between Wired, Free Space Wireless and Intra Chip Wireless Links

Metric	Wired	Free Space Wireless	Intra Chip Wireless
Latency (picosec)	1159.3	33.33	473.5
Energy (pJ/bit)	18.2517	0.612	4.575

5.9.1 Interconnect Latency

The packet latency while transmitting data using wireless links depends on SERDES delay, serial bit stream frequency and filtering delay along with the propagation delay of wireless channel. Table 5.3 shows the latency and energy values of buffered wired, free space wireless and intra chip wireless links at 20mm distance between source and destination nodes. The propagation delay in on-chip wireless channel is ≈ 473.5 picosec using directional antennas. The propagation delay with omnidirectional antennas is further lower than that with directional antennas. In comparison, propagation delay for free space wireless channel link, long unbuffered and buffered wired links is 33.33 picosec, 3860.2 picosec and 1159.3 picosec respectively. While latency using wireless links is significantly better than wired links, it can be noted that packet latency in WNoCs is tenfold higher as compared to that of free space channel. The high propagation delay increases the active time and channel access time for each transmission, leading to higher energy consumption and challenges for efficient management of shared channel among all wireless nodes.

Additionally, channel delay dispersion further increases overall packet latency to receive all signal compo-

nents at the receiver. Especially, omnidirectional links have dispersion running into several nanosec. Combining propagation delay and delay spread along with other circuit delays, the worst case latency with directional and omnidirectional antennas is 550picosec and 3400picosec (without any compensation) respectively. While directional links perform 85.75% and 52.56% better, omnidirectional links perform the same as compared to long unbuffered and buffered wired links, Considering wired link layout issues and transfer through multiple layers, on-chip wireless links provide significant benefits at long distances. However, data transmission latency with delay dispersion runs into several system clock cycles and needs to be accounted for designing and evaluating WNoCs.

5.9.2 Energy Consumption

The energy per bit for an interconnect gives a measure of energy expended in transmitting a bit and is the total sum of power dissipated in all wireless components for the entire duration of the transmission. All the OOK based transceiver components consume 32mW of power. At 20mm distance, path loss in on-chip wireless channel for directional and omnidirectional antennas is 39dB and 29dB respectively. Furthermore, omnidirectional links require high frequency bandpass filters [24] to compensate for high delay spread, which increases overall loss to 34dB. Considering all losses and power requirements, on-chip wireless links consume 4.575pJ/bit (for directional antenna) to transmit data over 20mm distance, as opposed to 0.612pJ/bit, assuming free space channel. The transceiver design, especially in terms of transmission power requirements needs to take into account the higher path loss of on-chip wireless links. In comparison, long wired links consume 18.2517pJ/bit and 82.0257pJ/bit energy at 20mm distance for buffered and unbuffered respectively [22]. In addition, energy consumption for wired links increases exponentially with distance, while wireless link energy increases linearly providing high energy savings for long distances across the chip.

5.9.3 Network Performance

The analysis of link latency and energy consumption provides an understanding of how on-chip wireless links perform as compared to free space channel and long wired links. We utilize this to measure the network performance with intra chip channel characteristics. WNoC topology used for the evaluation is adopted from [1] and augments wired mesh topology with mm-wave wireless links. The number of wireless links at different system sizes is 3, 7, 10 and 14 for 64, 256, 512 and 1024 cores respectively. Fig. 5.17 shows normalized packet latency and energy under uniform random traffic for different system sizes with wireless links for both free space and on-chip channel cases. The latency and energy values are normalized by corresponding values of

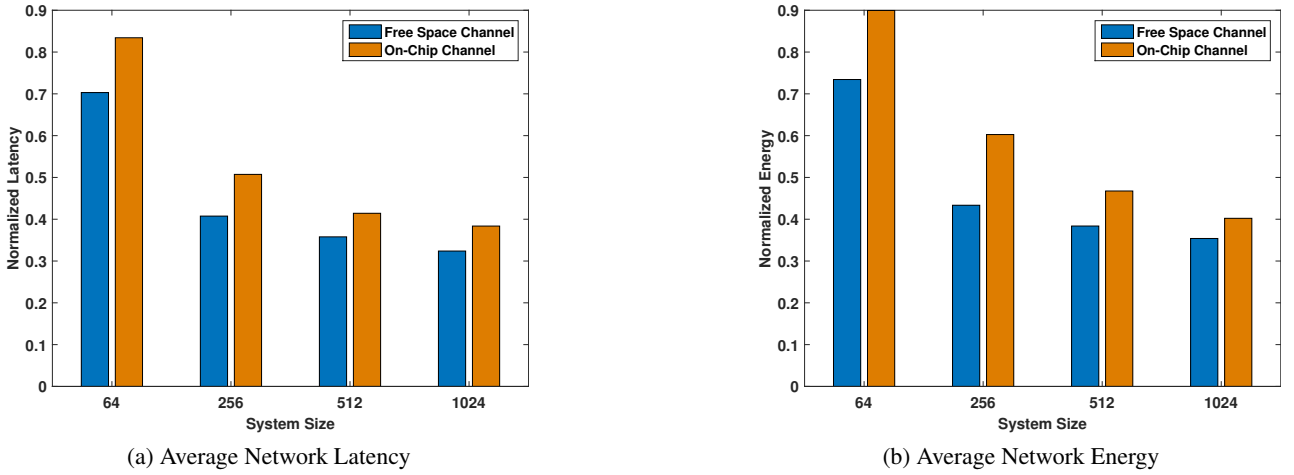


Figure 5.17: Comparison of Network Performance with On-Chip and Free Space Wireless Communication Models

baseline wired mesh topology of same system size.

As observed from Fig. 5.17, WNoC performs better than wired only mesh topology, but the amount of improvement attained is slightly lower as compared with that of free space assumption. The average packet latency with on-chip wireless links is higher by 15% – 25% than that of free space wireless links for different system sizes. For a system with 256 cores, WNoC provides 50.72% improvement in latency as opposed to 60% when free space channel is considered. In terms of packet energy, WNoC with on-chip wireless links offers little savings at smaller system sizes, unlike WNoC with free space channel assumption. For a 256-core system, WNoC saves 40% energy over wired mesh and the energy savings become more significant as system size increases further. This can be observed from the fact that, with increasing number of cores, distance between nodes increases and benefits of wireless links become more apparent. The energy savings can be further improved by use of power gating for transceiver circuits. The analysis of network performance at different system sizes shows that, though performance of on-chip wireless links is lower than free space links, WNoC offers considerable savings in latency and energy for many core architectures.

5.10 Impact on WNoC

The major takeaways from the analysis of on-chip wireless channel and impact of the antenna placement are - (i) on-chip wireless channel is highly fading in nature due to several propagation paths and (ii) antenna placement and directionality give rise to unique variations in signal strength and delay spread. While directional antennas provide better performance in terms of channel delay spread, antennas placed at chip corners are still severely

impacted. This mandates the incorporation of antenna characteristics and channel effects in the circuit and system level design and placement of transceivers in WNoCs. In addition, different transceivers experience different signal characteristics, requiring either careful tuning of each transceiver or design for the worse case. A careful analysis of all physical level parameters have be considered for both design and evaluation of WNoCs. Furthermore, the high delay spread results in small coherence bandwidth, leading to fading effects even at very low frequencies and necessitates fading resilient modulation schemes, bandpass filters at receivers, etc.

5.11 Design of Wireless Transceiver

Another major factor critical to the performance and overheads of the wireless links and network is the design of transceiver and modulation scheme chosen. On-Off Keying (OOK) modulation scheme has been widely adopted for energy efficient transceiver designs because of their low area and power overheads[141, 142, 143]. A fully integrated OOK transceiver operating at 60 GHz[141] achieves a maximum data rate of 10.7 Gbps with 67mW power consumption. An energy efficient OOK based transceiver is proposed in [142, 143], that reduces transceiver energy consumption by avoiding power hungry Phase Locked Loops (PLLs) and using a common Voltage Controlled Oscillator (VCO) between transmitter and receiver. They achieve data rate of 18.7 Gbps at 60 GHz with 0.25 pJ/bit energy consumption. Though OOK based transceivers have low area and power overheads, they have poor spectral efficiency and are susceptible to channel effects.

Analysis of intra chip wireless channel in section 5.1 shows that it varies significantly from free space characteristics and is significantly impacted by high frequency distortions, multipath propagation, channel fading, near field effects and is susceptible to antenna placement and specifications. They severely degrade the transmitted signal, resulting in either low performance or increased overheads due to complex filtering circuits. OOK modulated signal is susceptible to effects like channel fading and hence does not provide optimal performance in WNoCs. While traditional wireless systems can make use of error detection/correction techniques and retransmissions, such techniques in WNoCs can be cost ineffective. A few works have tried to achieve more robust and efficient on-chip wireless transceivers by use of other modulation schemes. A comparison between different shift keying schemes in [144] shows that Quadrature Phase Shift Keying (QPSK) achieves twice the bandwidth as compared to OOK modulation. A scalable WNoC architecture in [145], uses Code Division Multiple Access (CDMA) with Walsh Codes to achieve peak network bandwidth of 12 Tbps and packet energy of ≈ 250 nJ for a 256-core system. Furthermore, increasing bandwidth requirements of emerging applications (CPU, memory, graphics, etc.) require high data rate and spectrally efficient wireless links to sustain the performance demands. Therefore, design and reliability analysis of transceivers that are resilient to channel effects, provide

high data rate with low overheads is critical to achieve energy efficient, high performance WNoCs.

To design an efficient transceiver with channel resilience, we propose Orthogonal Frequency Division Multiplexing (OFDM) based transceiver operating at mm-wave frequencies. OFDM is a multi-carrier modulation technique, wherein a wide-band channel is divided into multiple sub-channels for improved reliability and spectral efficiency[146]. OFDM, by use of orthogonal sub-carriers, has the ability to perform well against Inter-Symbol Interference (ISI) and channel effects like frequency selective fading, high frequency attenuation, etc. In addition, OFDM technique is discrete implementation of multi-carrier modulation, which allows for it to be implemented mostly through digital design with low area overheads. The channel characteristics like path loss and delay spread derived from wireless channel modelling in section 5.6 are used to determine the specifications of OFDM transceiver. The major highlights of our OFDM based transceiver design and analysis are:

1. Design and implementation of OFDM transceiver, that is robust against channel effects and provides high data rate for WNoCs.
2. Detailed evaluation of data rate, BER, area and power consumption with different channel bandwidths, sub-channels and data mapping schemes.
3. Evaluation of network performance and overheads using OFDM scheme under different benchmarks.
4. Reliability analysis and comparison of OFDM and OOK based transceivers.

5.11.1 OFDM Transceiver for WNoCs

OFDM typically encodes data bits into parallel symbols, with each symbol modulated to one sub-channel of transmitted signal. The design of OFDM and its robustness against channel effects is primarily impacted by number of orthogonal sub-channels and encoding scheme employed.

5.11.1.1 Sub-channel Design Criteria

Though, theoretically any number of sub-channels can be used in OFDM scheme, it is decided by channel's coherence bandwidth (B_C). For a signal transmitted over wireless channel to experience relatively flat fading, channel bandwidth must be sufficiently smaller than its coherence bandwidth, $B_C = 1/T_m$.

As seen from Fig. 5.8, the highly multipath propagation of intra chip wireless channel results in high delay spread, especially for omnidirectional antennas. The delay spread varies from as low as zero up to as high

as $5ns$ depending on antenna design. OFDM scheme overcomes channel delay spread impacts by transmitting multiple parallel data streams at slower rate over the sub-channels. The low symbol rate also helps in mitigating ISI effects. To make the OFDM modulator robust, we consider design specifications to account for the worst case coherence bandwidth of $0.2GHz$ corresponding to $5ns$ delay spread observed in mm-wave frequencies. Hence, the number of orthogonal sub-channels is chosen to satisfy sub-channel bandwidth condition, $B_N = B/N \ll 0.2GHz$, where B is total bandwidth and N is number of sub-channels. The number of sub-channels, in general, is chosen to be power of 2 for easy implementation of DFT. For $B = 25GHz$, the minimum and optimal number of sub-channels is 128 and 256 respectively.

5.11.1.2 Cyclic Prefix

The multipath propagation of on-chip wireless channel can result in loss of sub-channel orthogonality, leading to ISI. OFDM inserts guard intervals in between symbols to eliminate this and improve the reliability of transmission. Cyclic prefix bits are transmitted during these guard intervals. It maintains sub-channel orthogonality by keeping the symbols periodic over the extended duration. They are the last μ signal samples of a symbol added in the guard interval period at the beginning of time domain signal sequence. To provide sufficient guard interval, the value of μ is determined from channel characteristics as $\mu = T_m/T_s$, where T_s is the sampling time of the signal. The cyclic prefix bits are discarded at the receiver and results in reduction of data rate by a factor of $N/(N + \mu)$. The power associated with these additional bits adds energy overhead to total transmission power. At the cost of slight increase in length of transmitted sequence and power consumption, cyclic prefix bits completely eliminate ISI and improve reliability of transmission.

5.11.2 Implementation of OFDM Transceiver

The OFDM transceiver implemented for WNoCs is shown in Fig. 5.18. OFDM design is broadly divided into three parts; (i) data encoding, (ii) implementation of DFT/inverse DFT and (iii) cyclic prefix and serial/parallel interfaces. We use M-QAM data encoding scheme to ensure high spectral efficiency with sufficient BER. It encodes $\log_2 M$ data bits per symbol into both amplitude and phase of the carrier signal. The FFT module modulates the encoded symbols to respective sub-channel frequency using inverse Discrete Fourier Transform (DFT). The resulting time domain signal is up converted to transmission carrier frequency (f_0) and amplified using Power Amplifier (PA) before transmission. The FFT radix depends on the number of sub-channels chosen. At the demodulator, received signal is down-converted and converted to parallel data streams. Finally, the cyclic prefix bits and symbols are demodulated using DFT operation of FFT module and decoding opera-

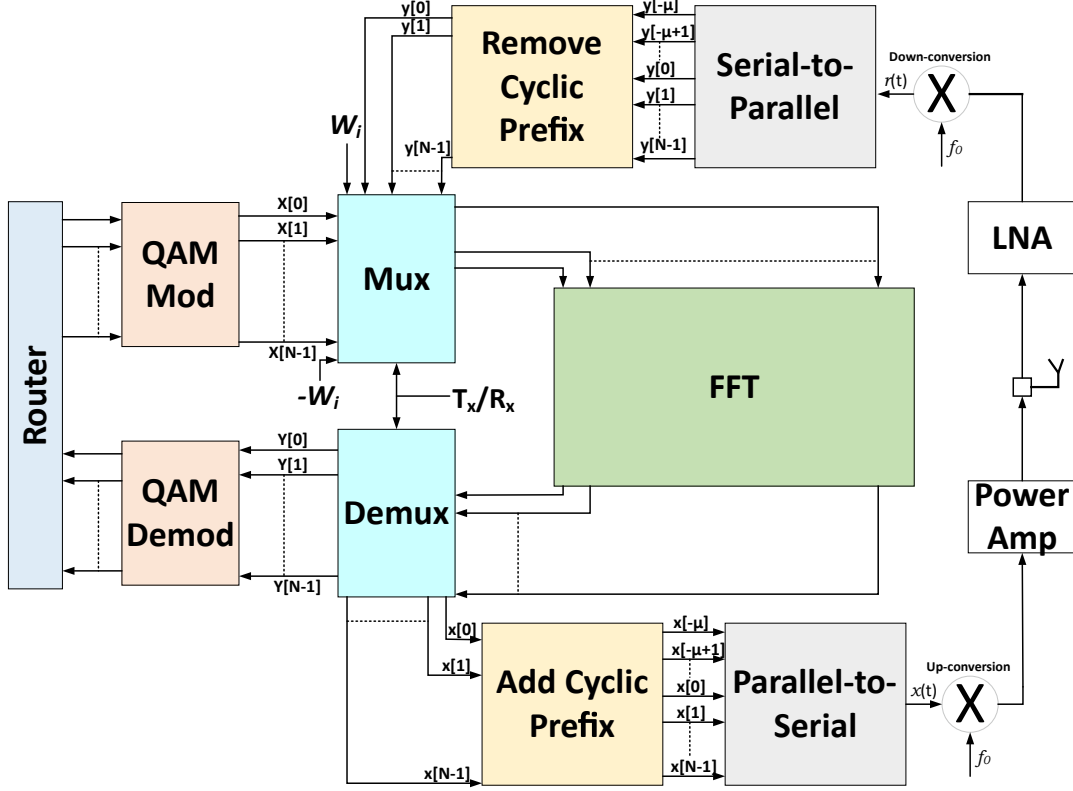


Figure 5.18: OFDM based Transceiver for WNoC

tions. The serial/parallel interfaces in the transceiver convert data between parallel data streams to time domain sequence.

In our implementation, the FFT module performing inverse DFT and DFT operations is reused between transmitter and receiver to reduce area overheads. Most WNoCs implement WIs with single antenna and single channel setups to keep the overheads to a minimum. Hence, only the transmitter or receiver is active at any time and so the transceiver has to either modulate or demodulate the signal accordingly. Utilizing this, we share the FFT hardware between modulation and demodulation operations of OFDM. The QAM symbols at modulation form the frequency components of OFDM signal. Inverse DFT at modulator converts these components to time samples and DFT at demodulator converts time samples back to corresponding symbols. The operations differ from each other only by the twiddle factor; $e^{\frac{j2\pi in}{T_N}}$ and $e^{-\frac{j2\pi in}{T_N}}$ for modulation and demodulation respectively. Hence, we use the same FFT hardware and provide appropriate set of twiddle factors for each operation to keep the area and power overhead to minimum. The T_X/R_X control signal chooses the appropriate factor by use of multiplexer and demultiplexer circuits as shown in Fig. 5.18.

5.11.3 Analysis of OFDM Transceiver

To evaluate the performance of OFDM scheme, we have implemented end-to-end communication with all channel characteristics in MATLAB. The OFDM transceiver is synthesized with Cadence RC compiler using $32nm$ technology to analyse its overheads. The designs for power amplifier and LNA are adopted from [142]. We have considered two antenna structures for all our evaluations, omnidirectional loop antenna[107] and directional, Planar Log Periodic Antenna (PLPA)[73] with $25GHz$ and $6GHz$ bandwidth respectively. The number of orthogonal sub-channels with worst case delay spread of $5ns$ is 256 and 64 for loop antenna and PLPA respectively. The number of cyclic prefix bits at worst case delay spread of $T_m = 5ns$ is 24 bits. Using MATLAB, we derive performance metrics of data rate, power consumption and Bit Error Rate (BER). The network performance is then evaluated in terms of application speedup, packet latency and packet energy by running different OpenCL[147] application benchmarks on Multi2sim[36] tool.

5.11.3.1 Data Rate

The data rate of OFDM scheme is computed with 64 and 256 sub-channels with 4 and 16 pilot signals respectively. The pilot signals in OFDM are inserted to estimate wireless channel characteristics at the receiver. We use 256-QAM as encoding scheme, which encodes $K = 8$ bits per symbol. The data rate, R of the OFDM scheme can be calculated as shown in (5.2),

$$R = K * (N_D + N_P + N_G) / (T_G + T_N) \quad (5.2)$$

$$T_N = 1/B_N; T_G = \mu/B$$

N_D , N_P and N_G are data sub-channels, pilot sub-channels and cyclic prefix. T_N is symbol duration and T_G is the cyclic prefix duration. With maximum delay spread, the data rate achieved is 43.6 Gbps and 195.32 Gbps for PLPA and loop antennas respectively. By using zero cyclic prefix for PLPA (zero delay spread from Fig. 5.8b), the data rate is improved to 47.4 Gbps. As expected, addition of cyclic prefix decreases the data rate achieved as it increases the length of time domain sequence. The implemented OFDM scheme achieves a maximum spectral efficiency of $7.2bps/Hz$, taking into account worst case delay spread. A comparison of proposed OFDM scheme with other existing transceiver designs in Table 5.4 shows that OFDM achieves high data rate with comparable overheads. Furthermore, it overcomes the impacts of wireless channel, unlike widely used OOK modulation.

Table 5.4: Performance of Proposed and Existing Transceivers for WNoCs

Modulation	Tech	Data Rate (Gbps)	Energy (pJ/bit)	Area (mm ²)
OOK[141]	90nm	10.7	6.26	0.44
OOK[142, 143]	65nm	18.7	1.2	0.12
OFDM	32nm	43.6	0.498	0.073
		195.32	0.132	0.092

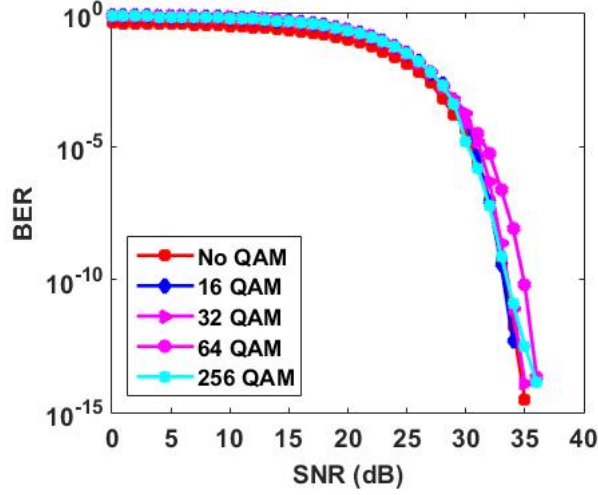


Figure 5.19: Variation of BER for different OFDM Specifications

5.11.3.2 Bit Error Rate

The bit error rate of the proposed scheme is evaluated by incorporating the intra chip wireless channel characteristics into the OFDM modulation scheme. The noise in the channel is modeled as additive white Gaussian noise. We transmit 10000 data frames (1 frame per cycle). The BER as variation of Signal-to-Noise Ratio (SNR) for different encoding schemes is presented in Fig. 5.19. The BER achieved at SNR of 35dB is in the order of 10^{-14} , which is traditional for wireless links.

5.11.3.3 Overheads

The total area consumed by all components with 64 and 256 sub-channels is $0.073mm^2$ and $0.092mm^2$ respectively. The power consumption in OFDM modulator/demodulator is $2.73mW$ and $6.825mW$ at 64 and 256 sub-channels. The power amplifier and LNA combined together have power consumption of $19mW$ [142]. Hence, the OFDM based transceiver consumes $0.498pJ/bit$ and $0.132pJ/bit$ at 43.6 Gbps and 195.32 Gbps data rate respectively. Table 5.4 shows a comparison of area and energy consumption of proposed OFDM scheme with other existing transceiver designs. The energy per bit with OFDM is significantly better as compared to any of the existing transceiver designs. Hence, the proposed scheme achieves significantly high data

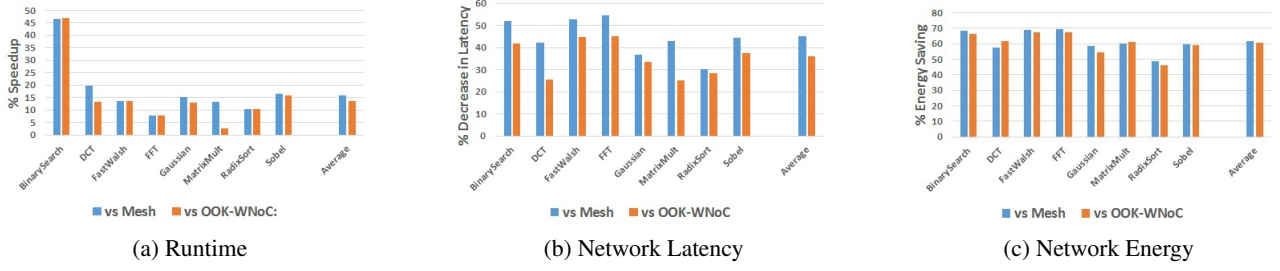


Figure 5.20: Performance and Energy Improvements using Proposed OFDM Scheme for Different Benchmarks

rates with comparable per bit energy consumption.

5.11.4 Evaluation of Network Performance

In this section, we evaluate the WNoC performance with the proposed OFDM modulation on a 64 core system. To find the placement of wireless transceivers, we have adopted the simulated annealing based placement optimization from [52]. We compare the performance with that of traditional wired mesh and OOK based WNoC topologies. The placement of transceivers in both OOK based and OFDM based WNoCs is kept same.

5.11.4.1 Application Runtime

Fig. 5.20a shows the application speedup provided by OFDM based WNoC over mesh and OOK based WNoC topologies. The values presented in the figure are averaged over ten runs of each application. On average, OFDM based WNoC speeds up the application by 15.9% and 13.5% over mesh and OOK based WNoC respectively. The runtime improvements are obtained by higher bandwidth provided by OFDM thereby resulting in fewer cycles for each packet transfer.

5.11.4.2 Network Latency

Fig. 5.20b shows the reduction in average packet latency by OFDM based WNoC over mesh and OOK based WNoC topologies. The latency includes both on-chip and off-chip access latency to obtain the data. On average, OFDM based WNoC reduces latency by 42.3% and 36% over mesh and OOK based WNoC respectively. The runtime improvements are obtained by higher bandwidth provided by OFDM thereby resulting in fewer cycles for each packet transfer.

5.11.4.3 Network Energy

Fig. 5.20c shows the energy savings provided by OFDM based WNoC over mesh and OOK based WNoC topologies. WNoC topologies significantly improves packet energy due low per bit energy of wireless links over long distances as compared to global wires. OFDM scheme further reduces total energy due to its low power efficient implementation using FFT. On average, OFDM based WNoC reduces the energy by 61.6% and 60.5% over mesh and OOK based WNoC respectively.

5.12 Robust Wireless Channel Access Mechanism

The improvements to antenna efficiency, channel characteristics and transceiver performance are further enhanced in our work through a robust channel access mechanism for WNoCs. The wireless channel sharing and allocation among different transceivers is one of the major challenges of WNoC implementations. On-chip wireless links, in general, adopt single channel communication to keep overheads to a minimum and hence require a mechanism to allocate and share the channel judiciously with each transceiver. Even with multi-channel implementation, it is not feasible to allocate separate channel for each transmission and the channels must be shared. Token passing mechanism[148, 149, 150, 151] has been adopted to provide access of wireless channel and to eliminate the overheads associated with centralized controllers. Centralized access control mechanisms have access to global information about all transceivers, but incur significant overheads to transfer control signals.

Token passing control performs efficiently as compared to centralized mechanisms, but there are still several issues with their current implementations. Most existing implementations of token passing use a timer based approach[149, 151] to determine passing of token between transceivers. Though this approach is simple, it is not flexible to accommodate different requirements of WNoCs. Firstly, data packets, even in homogeneous architectures, vary in the number of flits and secondly, the load at each transceiver varies considerably with time. Hence, a fixed timer based token passing does not efficiently accommodate the varying packet size and transceiver utilization scenarios, leading to inefficient use of the channel. This further becomes a critical limitation in WNoCs for heterogeneous system architectures like HyWin. In heterogeneous architectures, different computing subsystems have different latency and bandwidth requirements, leading to an even skewed WI channel access requirement and a simple timer based access control method does not cater to this.

To overcome the limitations of existing token passing mechanisms, we propose a comparator based controller design that performs channel allocation to different transceivers. The comparator in the proposed con-

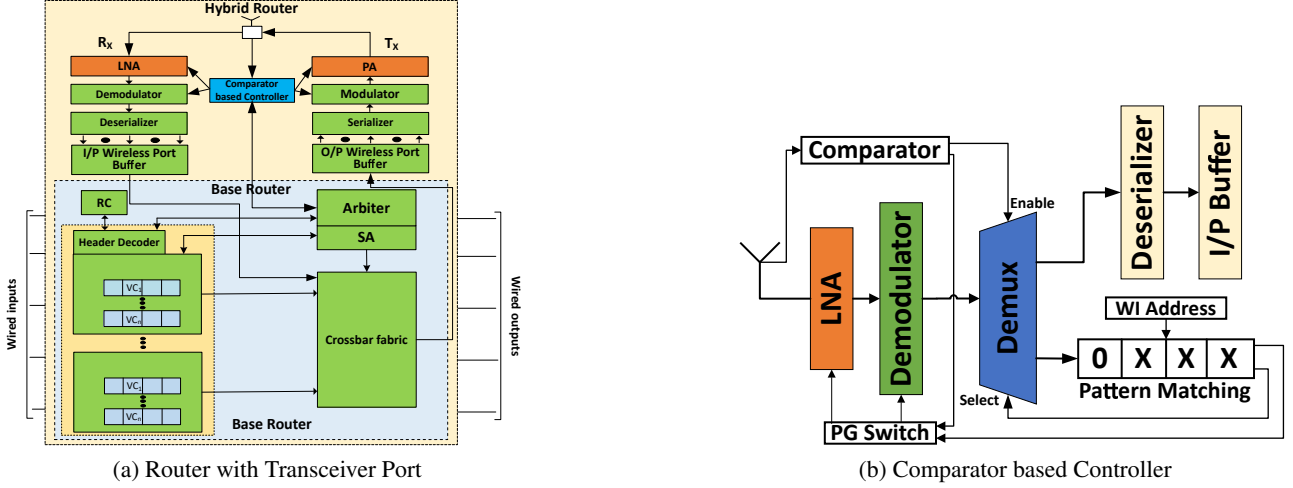


Figure 5.21: Comparator based Controller for Channel Access and Power Gating in WNoCs

troller is attached to the antenna in transceiver and pre-processes the transmitted data to control channel allocation. In addition to channel allocation, the comparator based controller also controls the power gating operation to reduce the power consumption in wireless transceivers. Transceivers, not actively involved in transmission, can be power gated to significantly reduce their leakage energy [14][15][16]. For channel allocation, we append additional information in the packet header transmitted over wireless links. This information and comparator based controller, not only perform channel allocation, but also help set additional features like channel access time, channel access priority, making the proposed mechanism robust and applicable to most scenarios.

5.12.1 Comparator based Controller

The comparator based controller design for performing channel allocation and hybrid router with wireless transceiver are shown in Fig. 5.21. The comparator is attached to the antenna and processes the wireless signal as it is received over the channel. A pattern matching circuit sits at the output of demodulator, which demodulates the signal and converts it to serial digital data stream before it is deserialized and written to the router input buffers. A Demux circuit controls the data stream to switch between the pattern matching circuit and input buffers.

To perform channel allocation and determine appropriate receiver for a transmission and arbitration control, we modify the header flit of the packet transmitted over the wireless link. The modified packet stream is shown in Fig. 5.22. The dummy bits added at the start of packet transmission are to compensate for wake-up latency associated with power gating. For channel allocation, a unique address (*WI Address*) is assigned to each transceiver and the *WI Address* of both source and destination transceivers are added to the header flit. These addresses are different from the actual source and destination addresses of the packet, which are also

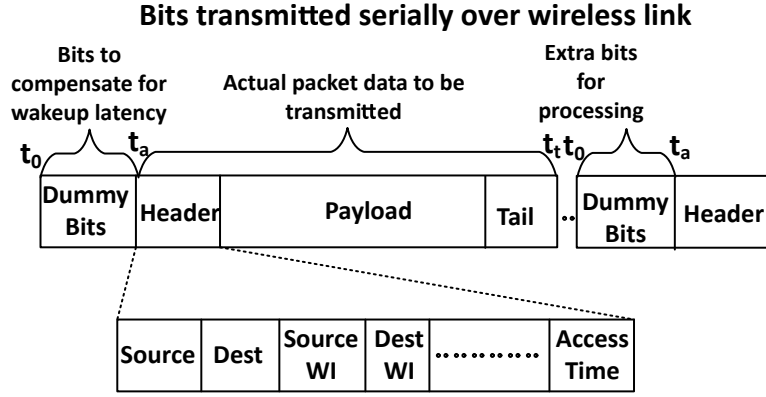


Figure 5.22: Format of Packet Transmitted over Wireless Link for Comparator based Channel Allocation

present in the header flit. The source *WI Address* is used to determine the arbitration and token passing, while the destination *WI Address* is used to resolve packet destination once transmission starts.

Once a transceiver initiates a transmission, the transmitted signal is received at all transceivers and the associated comparator detects a valid received signal. Upon the detection of valid signal and receiving of header flit, the source and destination *WI Address* bits are processed by pattern matching circuit. At each transceiver, the received source *WI Address* is matched with the address of transceiver that comes before itself in the arbitration. If there is a mismatch, the source address bits are discarded. Otherwise, the controller sets a single bit token signal to indicate that the transceiver is the next in arbitration to get access of the channel, thereby passing the token. The elegance of the proposed method is that it passes the token with few modifications to the packet format. It does not require token to be passed separately or incur additional control signalling overheads. The destination *WI Address* is matched with its own address to verify if the intended receiver of the packet is the current transceiver or not. If the address matches, the packet reception is continued, otherwise the reception is stopped using power gating mechanism. The output of the comparator indicates the start and end of any transmission, thereby allowing the next transceiver in arbitration to start its transmission immediately.

To ensure fair usage and allocation of channel among all transceivers, the access of channel to each transceiver is controlled based on load at the transceiver. The total packet load to be transmitted at each transceiver is divided into multiple levels in the proposed method and channel access time (time for which the channel is allocated to the transceiver) is set proportional to the current load at the input buffers of transceiver. The channel access time in proportion to input load is universal across all transceivers and the proposed method ensures that channel allocation is efficient in accordance with the utilization of each transceiver in the network. The channel access time and arbitration sequence are both written in user controlled buffers at each transceiver, thereby allowing them to be modified as per the user needs. Furthermore, the channel access time for each transmission is also included in the header flit as shown in Fig. 5.22, allowing the receiving transceivers to

make adaptive routing decision in regards with the usage of wireless links. By using simple comparator based controller and few modifications to the header flit format, the proposed method performs channel allocation, token passing and control of channel access time and arbitration sequence providing greater flexibility to match the requirements of any application or system architecture.

5.12.2 Transceiver Power Gating Control

The comparator based controller, in addition to performing channel allocation, also reduces energy consumption in transceivers by power gating them when they are not actively involved in communication. Initially, all transceivers are kept in idle state. At any hybrid router with wireless transceiver, when routing strategy decides to use transceiver and associated transceiver has token to transmit data, the transmitter is brought to active state. The transmitted signal from source transceiver is received at the antennas of all other transceivers and their receiving end is brought to active state upon detecting a valid signal in the channel. The comparator of the proposed controller differentiates the valid signal from channel noise. The threshold of the comparator is set to the average value of signal strength of the most far apart nodes and worst case noise floor value of the wireless channel. This allows the receiving transceivers to be brought to active state without any additional control information. The received packet is then demodulated and deserialized through transceiver components. A small wake up latency is associated with power gating mechanism and the data received over wireless channel are lost while the transceivers are brought to active state. To avoid this, dummy bits are added at the start of packet and the size of dummy bits is proportional to the wakeup latency of the power gating switch. Furthermore, since all transceivers, along with the intended receiver of the packet become active, the packet is unnecessarily processed at all transceivers, leading to unwanted energy consumption. The pattern matching circuit of received *WI Address* is used to detect the intended receiver of the packet (as described before) and all transceivers (except the actual receiver) are put back in sleep state once the pattern matching operation is completed. This reduces the unwanted data processing and energy consumption at all transceivers, thereby further increasing the energy efficiency of proposed method. While the transmission is active, all the components in both transmitting and receiving transceivers are kept active. Once the transmission is complete, the source transceiver relinquishes the access of channel, both transceivers are put back to sleep state and next transceiver in arbitration starts transmission as described before. Hence the proposed comparator based design also controls power gating of transceivers efficiently with little overhead to reduce the energy consumption.

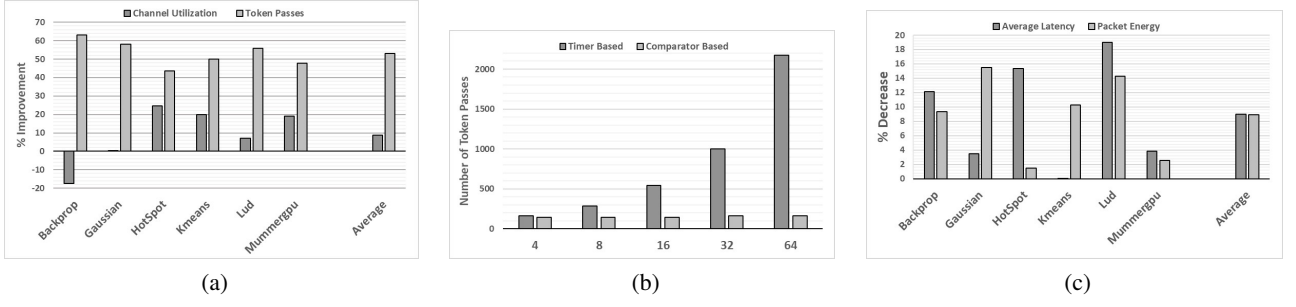


Figure 5.23: (a) Improvement in Wireless Channel Utilization and Token Passes (b) Average Number of Token Passes across all Transceivers for Different Packet Sizes and (c) Decrease in Average Packet Latency and Energy Consumption using Comparator based Controller

5.12.3 Performance of Comparator based Controller

We analyse the efficiency of proposed comparator based controller using HyWin topology and Rodinia suite[79] on Noxim[152] network simulator. It is evaluated in terms of channel utilization, token passes, packet latency and energy consumption. The results are compared with timer based token passing mechanism, which uses 8 cycle access time for each transceiver in one arbitration sequence.

5.12.3.1 Wireless Channel Utilization

We first evaluate the efficiency of proposed controller in improving channel utilization and allocating the channel to each transceiver. Fig. 5.23a shows the improvement in channel utilization and channel allocation over the entire application duration for different benchmarks. The channel allocation improvements are measured in terms of the number of token passes at each transceiver. As observed from the figure, comparator based controller improves the wireless channel utilization over most of the benchmarks. On average, it improves channel utilization by 9% as compared to timer based approach.

Similarly, the number of token passes at each transceiver using comparator based approach is reduced by 53%, on average over timer based token passing mechanism. Unlike a timer based approach which uses fixed duration at each pass, proposed approach sets the channel access time as per the load during that pass. This makes the token passing adaptable to the packet sizes and utilization of each transceiver, thereby improving both channel utilization and token passes. This is evident from Fig. 5.23b, which shows the number of token passes with different packet sizes for Backprop benchmark. In this scenario, the packet size is kept fixed in each case. As the packet size increases, number of token passes using timer based approach increases exponentially. At large packet sizes, each transceiver can transmit only a part of the packet in fixed time duration and hence requiring multiple token passes for each packet. This significantly increases the total number of token passes.

In comparator based approach, the channel access time is proportional to transceiver input load. Hence, a large packet size translates to larger access time, thereby keeping the number of token passes constant at all packet sizes. On the other hand, if packet size is small such that time required to transmit is smaller than fixed timer, channel remains idle during the remaining period. Comparator based approach, in this case, keeps channel access time smaller and improves overall utilization of wireless channel as evident from Fig. 5.23a. This makes the channel based approach robust against different packet sizes, packet formats and requirements at each transceiver (as in heterogeneous systems).

5.12.3.2 Network Performance

To evaluate network performance, we compare the average packet delay of comparator based controller with that of timer based controller. Fig. 5.23c shows decrease in average latency using comparator based controller for different benchmarks. From the figure, it can be observed that comparator based approach has low average latency as compared to that of timer based WNoC, though both have same hardware topology. On average, proposed approach reduces latency by 9% as compared to timer based approach. The improvements in packet latency are primarily achieved due to the efficient allocation of channel access time in proportion to the input buffer load at each transceiver. By allowing larger access time to transceivers with high load or vice-versa, the proposed method allows for faster access to data thereby significantly reducing the overall latency of the network. This is especially the case with transceiver associated with GPU cores, which require high volume of data to be transmitted between the cores and main memory.

5.12.3.3 Energy Saving

Fig. 5.23c shows average packet energy savings using comparator based WNoC as compared to timer based WNoC. In both implementations, power gating is employed for transceivers to save energy. As can be observed from the figure, comparator based WNoC reduces packet energy consumption by 9%, on average as compared to the timer based approach. Even though both approaches employ power gating, comparator based approach controls the transceivers in accordance with their load, thereby achieving better utilization of transceivers. This directly impacts the energy consumption of packets transmitted over wireless links, leading to further decrease in transceiver energy with comparator based approach as compared to timer based approach.

5.13 Concurrent Transmission Enabled Wireless Network Design

The OFDM based transceiver provides reliable and high data rate wireless network at low area and energy overheads. We further improve the performance of OFDM based wireless network by using the inherent sub-channel division mechanism to enable concurrent wireless communications without employing multiple frequencies.

5.13.1 Concurrent Transmissions using Sub-Channel Grouping

One of the key limitations of most existing WNoC topologies is single active wireless transmission at any given time due to the use of single channel communication. This limits the performance gains, utilization of wireless resources and increases idle energy consumption inactive transceivers. While multi-channel wireless[74] design provides simultaneous transmissions between the transceivers, it adds high overheads for achieving interference free communication over non-overlapping channels. Hence, alternative approaches are required to provide reliable, concurrent wireless transmissions, while also adding low overheads. In the proposed design, we achieve concurrent communication between multiple transceivers by grouping multiple sub-channels of OFDM scheme together and allocating one group to each transceiver. The transceiver transmits data only over the designated group of sub-channels, while leaving the other sub-channels unmodulated. We make minor modifications to the OFDM transceiver from Fig. 5.18 to allow transmission over single sub-channel group and reception over all other groups. Since, the sub-channel division is inherent to OFDM, the proposed concurrent communication methodology does not require multiple transceivers and adds very little overhead to the OFDM transceiver and WNoC topology. This allows different transceivers to transmit data simultaneously without interference over single channel. Furthermore, it retains all the advantages of OFDM viz., resilience to channel dispersion, ISI and high spectral density. Though each transceiver has access to only a portion of channel bandwidth, the total bandwidth of the wireless network remains the same for given implementation.

Since, each wireless transceiver has access to only a portion of bandwidth, the grouping of sub-channels and their allocation to each subsystem plays a pivotal role in meeting its bandwidth requirements. In systems with similarly configured subsystems, equal division of available sub-channels into K groups, where K is number of wireless transceivers, provides a simple and fair allocation of available bandwidth. But, such an allocation mechanism is not efficient for heterogeneous architectures, especially one with contrasting architectures like CPU and GPU on a single chip. Hence, to achieve a more optimal allocation of the sub-channel bandwidth, we utilize the average and peak bandwidth of data transfers (across several applications) of different computing clusters in a given architecture. The number of sub-channels assigned to each group is determined in

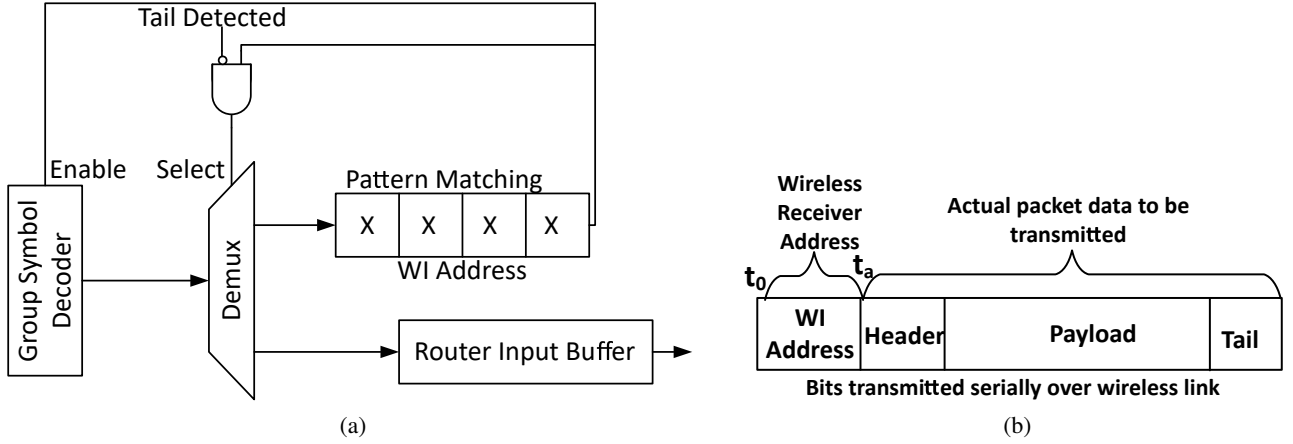


Figure 5.24: (a) Pattern Matching Circuit to Detect Packet Receiver (b) Data Format Transmitted over Sub-Channel Group

proportional to the each computing cluster's bandwidth as

$$N_i = \frac{B_i}{\sum_{i=1}^K B_i} * N_T \quad (5.3)$$

N_i is number of sub-channel assigned to transceiver i , N_T is total number of sub-channels and B_i is the bandwidth desired by computing cluster i . Simulation of several benchmarks on a heterogeneous CPU/GPU architecture has shown a recurring trend, where main memory requires the highest bandwidth, followed by GPU cluster and finally the CPU cluster. This matches the expected trend as GPUs are highly throughput sensitive, while CPU desire low latency communication for its mostly serial execution.

5.13.2 Data Reception and Decoding

The sub-channel grouping and allocation to individual transceivers allows all of them to transmit data simultaneously without interfering with each other. At the receiver end, the data from different transceivers transmitted over different sub-channel groups is received as combined input at the demodulator. Hence, the received signal needs to be appropriately disaggregated and decoded to obtain the data from different transceivers. This is achieved through a combination of OFDM symbol decoders and pattern matching circuits for individual groups as shown in Fig. 5.24.

The packet data transmitted by each transceiver is formatted as shown in Fig. 5.24b, which adds *Wireless Address* bits at the head of the actual packet. The *Wireless Address* is a unique identifier assigned to each transceiver and helps reduce the data processing and time required to determine if the received packet is intended for the same receiver. At the receiver, the received bits are decoded for this *Wireless Address* using a

pattern matching circuit as shown in Fig. 5.24a. Once the *Wireless Address* has been verified, the demux select line is chosen to send the received data to buffers, if the address matches. If there is a mismatch, the packet reception and decoding for that sub-channel group is ceased to avoid unnecessary energy consumption. The pattern matching decoder is operated at serial data stream frequency. The number of bits required for *Wireless Address* is as shown in (5.4), where N_{WI} is number of transceivers. Since, N_{WI} is small, the number of *Wireless Address* bits is very small. The maximum time taken to process these bits and stopping data reception, in case of mismatch, is equivalent to r bit cycles. Though it adds bit overhead to packet transmission, it ultimately avoids excessive energy consumption with a little control overhead. The total time taken to complete any packet transmission is given by (5.5), where N_{flits} is number of flits in the packet, S_{flit} is flit size, t_{bit} is serial bit duration and $t_{propagation}$ is propagation delay of wireless channel.

$$r = \log_2(N_{WI} + 1) \quad (5.4)$$

$$T_{max} = (r + N_{flits} * S_{flit})t_{bit} + t_{propagation} \quad (5.5)$$

5.13.3 OFDM Transceiver for Concurrent Transmission

The OFDM transceiver with sub-channel grouping methodology is shown in Fig. 5.25. In comparison with the OFDM transceiver in Fig. 5.18, it consists additional hardware for data encoding into respective group

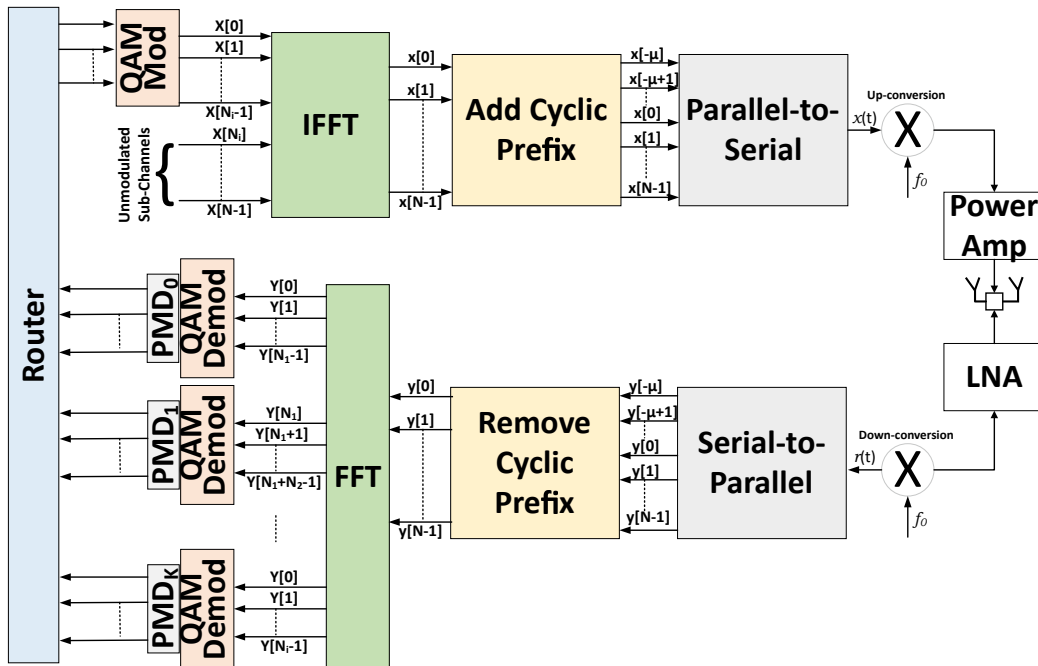


Figure 5.25: OFDM based Transceiver for Concurrent Transmissions in WNoC

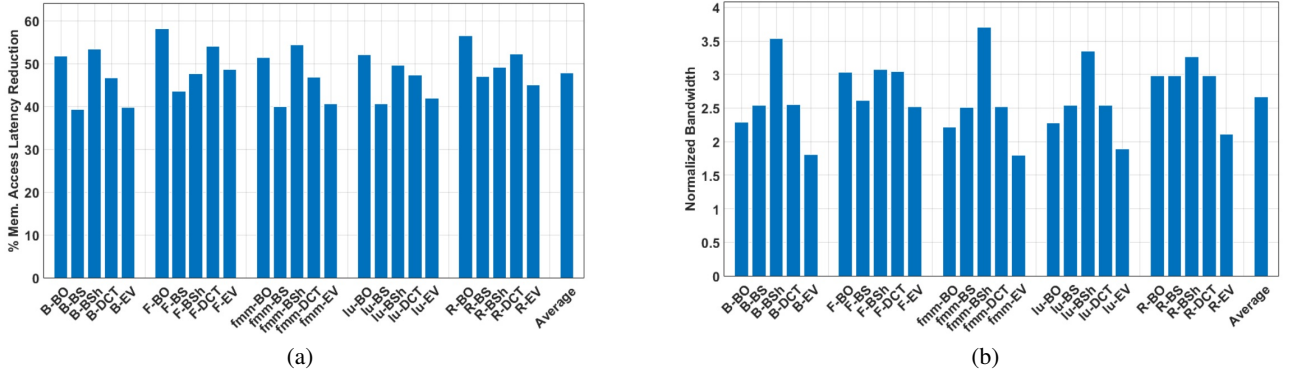


Figure 5.26: (Improvement in (a) Memory Access Latency and (b) Network Bandwidth using WNoC with Concurrent Transmissions over Wired Mesh

sub-channels at the transmitter and pattern matching and group symbol decoding on the receiver side.

As seen in Fig. 5.25, each WI encodes data only into its sub-channel group. At the receiving end, individual symbols are extracted from sub-channel components using FFT module after removing cyclic prefix bits. These extracted symbols are then fed into K pattern matching circuits corresponding to the K sub-channel groups. The initial *Wireless Address* bits are decoded as described before to determine the packet receiver and QAM decoding is continued, if packet receiver is a match. The output from each group are written to router buffers for further processing.

5.13.4 WNoC Performance with Concurrent Transmissions

The performance of OFDM transceiver with concurrent transmissions is evaluated on HyWin topology for a 64 core system using Multi2Sim[36] simulator and OpenCL[38] and SPLASH2[77] suites. The wireless links are implemented using loop antenna[107], which has total bandwidth of $25GHz$ at $58GHz$ carrier frequency. The 256 OFDM sub-channels are divided into 20%, 40% and 40% for CPU, GPU and memory subsystems respectively. The baseline topology is wired mesh with different cores mapped to provide non-overlapping paths to memory controllers as described in [19].

5.13.4.1 Network Performance

Firstly, we evaluate the network performance by comparing memory access latency and peak bandwidth with those of baseline mesh topology. Memory access latency is a major performance bottlenecks for processor performance and is one of the primary sources of contention in heterogeneous architectures. Fig. 5.26a shows an average reduction of 47.87% across different benchmarks in memory access latency using wireless network

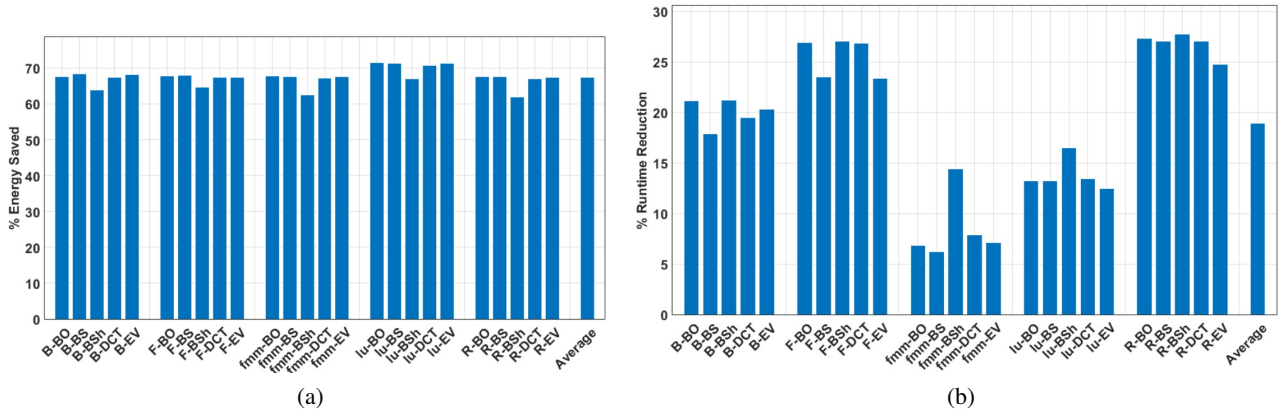


Figure 5.27: (Improvement in (a) Network Energy and (b) Application Runtime using WNoC with Concurrent Transmissions over Wired Mesh

with concurrent transmissions. This significant reduction is achieved by providing interference free transmission paths between CPU/GPU and main memory using sub-channel grouping, along with low latency of wireless links.

The improvement in peak network bandwidth is shown in Fig. 5.26b and shows the maximum throughput provided by enabling concurrent transmissions in WNoCs. On average, it provides $2.7\times$ more bandwidth as compared to baseline mesh. The improvements in memory access latency and network bandwidth clearly highlight the advantages provided by sub-channel grouping in providing concurrent communication between different subsystems.

Fig. 5.27a shows the savings in average packet energy with concurrent transmission enabled WNoC over baseline mesh. As observed, it saves 67.49% of network energy in comparison with baseline mesh.

5.13.4.2 Application Runtime

The application runtime improvement for different benchmarks are shown in Fig. 5.27b. The proposed WNoC topology with OFDM wireless links and sub-channel grouping provides an average runtime reduction of 18.79% as compared to baseline mesh topology. The presence of simultaneously applications running on both CPU and GPU leads to contention for network resources in mesh topology, leading to performance degradation. In the proposed topology, the heterogeneous subsystem networks reduce contention between different architectures and wireless links provide interference free, high bandwidth communication, thereby significantly improving the performance.

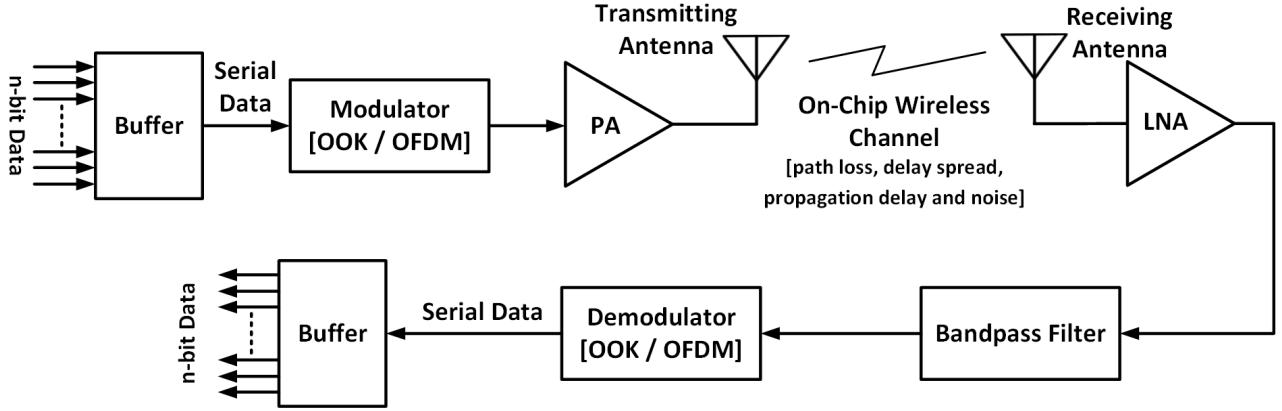


Figure 5.28: End-to-End Implementation of On-Chip Wireless Link for Reliability Analysis

5.14 Reliability Analysis of Wireless Links

The complex intra chip environment is the major contributing factor for deviation of on-chip wireless channel characteristics from free space channel. The adverse channel effects like multipath propagation, near field effects, etc. impact the reliability of transmission over the on-chip wireless links. In order to facilitate an understanding of wireless link reliability and channel performance in WNoCs, we implement on-chip wireless link end-to-end from data encoding at transmitter to data decoding at the receiver in MATLAB. As highlighted before, on-chip wireless channel exhibits high path loss, delay and dispersion characteristics further degrading the reliability of the data transmission in WNoCs. We consider OOK modulation scheme along with both OFDM modulation and data encoding schemes described in previous sections to provide well rounded reliability analysis in terms of Bit/Symbol Error Rate (BER/SER) and Power Spectral Density (PSD). Using the reliability analysis, we evaluate the impact on WNoC and provide design guidelines for improving the performance.

5.14.1 On-Chip Wireless Link Setup for Reliability Analysis

A typical process for transmitting data from output port at transmitting WI to the input port at the receiving WI, as shown in Fig. 5.28 involves (i) encoding data bits into symbols; (ii) modulating data symbols to carrier frequency; (iii) amplification and signal transmission through on-chip wireless channel; (iv) amplification and demodulation at the receiver and (v) decoding the symbols to obtain transmitted data. Each of these steps is modelled and reliability metrics like BER/SER and PSD are derived and compared against free space wireless links.

The efficiency and reliability of data transmission in WNoCs is largely dependent upon on-chip wireless

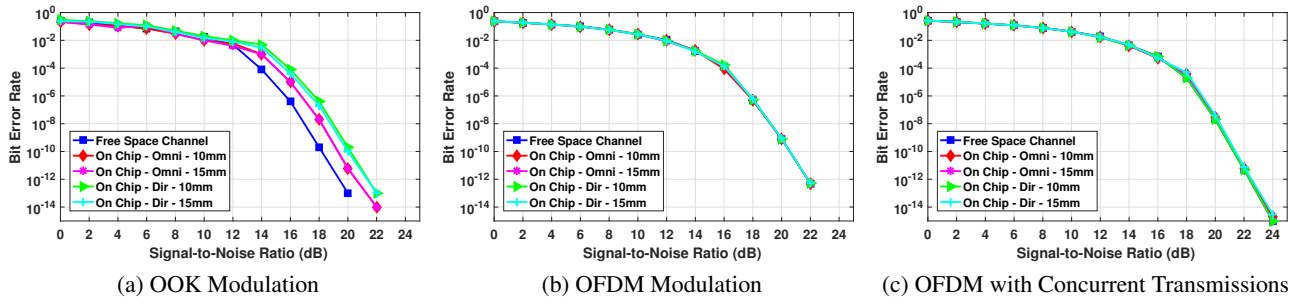


Figure 5.29: Comparison of Bit Error Rate of Free Space and On-Chip Wireless Channels for Different Modulation Schemes

channel characteristics like path loss, delay spread, delay, etc. The choice of data encoding and modulation schemes provides robustness against adverse channel effects to improve transmission reliability of wireless links. Furthermore, wireless network design specifications like number of channels, antenna properties, etc. impact and conversely impacted by desired reliability performance of on-chip wireless transmission. We utilize the detailed understanding of on-chip wireless channel and its characteristics derived in previous sections with different modulation schemes to analyse link reliability taking into account all these effects.

The OOK modulation scheme used in our reliability evaluation is adopted from [153, 154], which implements a non-coherent transceiver using OOK technique. The modulation circuitry consists of an up-conversion mixer, along with a power amplifier. The modulator impresses the input data onto the carrier generated by Voltage Controlled Oscillator (VCO). For data symbol 1, the modulated signal is set to the carrier signal, and is completely suppressed for data symbol 0. On the receiver side, direct-conversion topology is adopted, consisting of a low noise amplifier, a down-conversion mixer and a baseband amplifier. The injection-lock VCO is reused for both modulation and demodulation to reduce overheads. Despite of its implementation advantages, OOK does not ensure the data reliability as it suffers from the channel fading and dispersion.

5.14.2 Reliability Analysis

In this section, we evaluate the reliability of on-chip wireless links using the setup described before for different modulation schemes. We use BER and PSD of the wireless transmissions for understanding link reliability. The derived reliability metrics are used for analysing the impact on network performance and formulating guidelines for improving the transmission characteristics of on-chip wireless interconnects. The evaluated reliability and network performance is compared with those of free space channel assumption to understand the impacts of on-chip wireless channel.

Table 2.1 shows the chip specifications and antenna setup used for modelling the on-chip wireless channel

as described in section 5.1. The dimensions of the chip are set to $15mm \times 15mm$. The antennas are etched in the top metal layer and are designed to operate at a frequency of $60GHz$. Both transmitting and receiving antennas are either omnidirectional or directional in nature. In directional setup, the antennas are placed to be in the end-fire region of each other. The distance between both transmitting and receiving antennas is varied and path loss, propagation delay and delay spread metrics at each distance are derived. To summarize the findings of channel model, the on-chip wireless channel is characterized by high loss and delay spread as compared to free space channel. While omnidirectional antennas have better loss variation, they are plagued by significantly high delay spread (in the order of nanosec) due to several reflections off of different components. Conversely, directional antennas have almost zero delay spread, but have high path loss for most distances. We use the derived metrics at $10mm$ and $15mm$ for both the distances in our reliability analysis.

5.14.2.1 Bit Error Rate

The first metric we evaluate is the BER of both free space and on-chip wireless channels for different modulation techniques. Multipath channels, in general, perform poorly in terms of BER due to effects like delay spread, ISI, Doppler, etc. Fig. 5.29 shows the average BER achieved for different channel and modulation setups. In our analysis, we primarily consider delay spread and ignore Doppler effects as transmitter and receiver are stationary in WNoCs. As expected, it can be noted from Fig. 5.29 that OFDM modulation scheme performs well for both free space and on-chip wireless channel models in terms of BER as compared to OOK modulation. OFDM scheme is specially designed to overcome the effects of multipath fading channels and this highlights its advantages in WNoCs for reliable data transmission. Traditionally, a BER of the order 10^{-12} or lower is considered reliable for data communication through wired interconnects in CMOS chips. Using the same metric, on-chip wireless channel requires $2dB$ more power for achieving the same BER as compared to free space channel for OOK modulation. This results in at least 15% increase in transmission power requirements for transmitting data reliably across WNoC. For OFDM, the additional power required is less than $0.5dB$ between on-chip and free space wireless channels. Furthermore, prevalence of highly multipath channel requires additional hardware to band-limit the received signal and further increases receiver power requirements for reliable data reception in the same vein as wired links.

5.14.2.2 Power Spectral Density

Another important measure for reliable wireless communication is the PSD and interference in adjacent channels. To minimize this effect, the power radiated to the adjacent band should be considerably lower than the

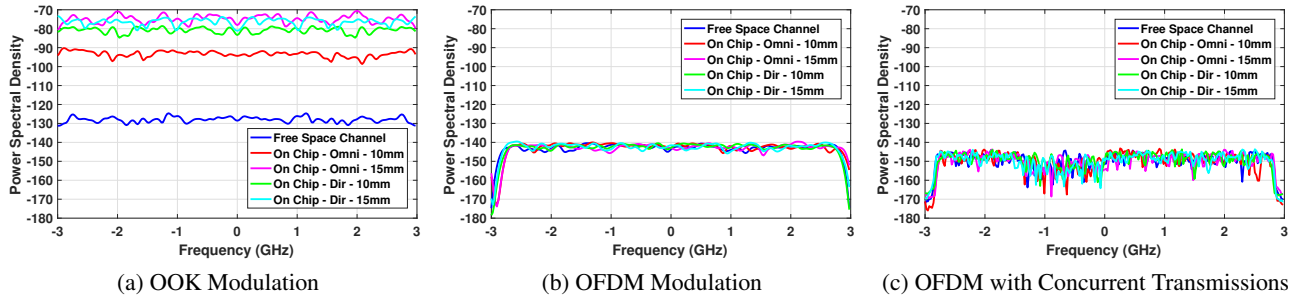


Figure 5.30: Comparison of Power Spectral Density of Free Space and On-Chip Wireless Channels for Different Modulation Schemes

power density of the desired band. Power spectrum is expected to have rapid decay at the boundary of the main band, otherwise the spectral regrowth into the side bands causes interference. Fig. 5.30 shows the power spectral density for all modulation schemes with free space and on-chip channel with different setups. The PSD is adjusted to be centered around the operating frequency of $60GHz$. It can be observed from the figure that the power spectrum for all approaches remains uniform across the desired band of frequencies. But, the PSD for OOK scheme remains high even at boundary frequencies, showing that considerable power can seep into adjacent bands. This nature of OOK makes it more prone to adjacent channel interference. On the other hand, for both OFDM approaches, the power spectrum decays rapidly at the edges indicating that the entire power is concentrated in the desired frequency band. In OFDM with sub-channel grouping, the PSD shows slightly erratic nature within the bandwidth of the channel showing some power being shared between different sub-channel groups. But the radiated power into other groups is not significant as indicated from the BER results. Overall, the OFDM scheme shows reliable performance even in presence of different adverse effects in on-chip wireless channel as compared to schemes like OOK.

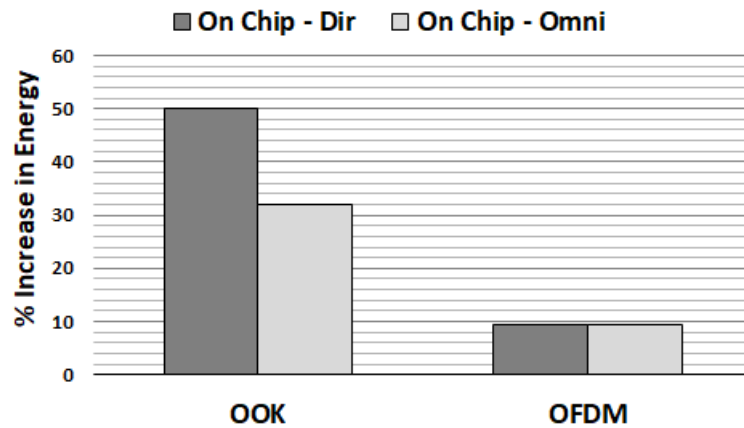


Figure 5.31: Increase in Average Packet Energy due to higher Transmission Power Requirement in On-Chip Wireless Channel

5.14.3 Impact on WNoC

The end-to-end link analysis of on-chip wireless channel shows that it has high power requirements to achieve reliable communication depending on the modulation scheme used. Besides the need for higher transmission power to maintain desired BER, receiver power requirements can be higher for band-limiting time dispersed signal and reliable data decoding. This is especially true for more simple modulation techniques like OOK. Comparing the two, while OOK has lower base overheads than OFDM, the net power overheads might be higher owing to higher BER, susceptibility to channel fading and ISI, etc. This further highlights the need for detailed evaluation and exploration of transceiver designs that perform well in on-chip wireless channel.

To illustrate the reliability impact of on-chip wireless channel, we show the increase in packet energy in intra chip environments as compared to free space channel in Fig. 5.31. The energy results are obtained using uniform random traffic in a 256 node NoC with 6 wireless transceivers and both directional and omnidirectional antennas. The transmission power for wireless link is set to achieve 10^{-12} with both channels for both modulation approaches. As can be observed from the figure, packet energy with on-chip wireless channel is upto 50% higher than that of free space channel when using OOK modulation. On the other hand, OFDM performs significantly better even in on-chip wireless channel with less than 10% increase in average packet energy. The evaluation presented here still does not consider any increase in overall latency of transmission or increase in receiver power requirements due to time dispersed signal. These affects can further increase the packet energy of data transmission in on-chip wireless link.

5.15 Conclusions and Future Work

Wireless links and infrastructure form the backbone to interconnection architectures proposed in this thesis and their proper understanding and efficient design is key to realizing desired performance. Towards this end, in this chapter, a comprehensive analysis of intra chip wireless channel and its impact on wireless NoC design using 3D model that captures key chip components influencing wireless transmission is presented. Design guidelines for antenna choice, location and implementation are provided to improve performance. Radiation efficiency and antenna bandwidth are greatly improved by placing antenna over a dielectric filled cavity. Analysis show that propagation in on-chip wireless channel is predominantly in near field/transition region, making channel more complex. Path loss and delay spread of the channel have been analysed and their inter-dependencies have been studied. Directional antennas are less effected by channel time dispersion but exhibit higher losses and vice-versa for omnidirectional antennas.

Utilizing these insights, we design a reliable, high bandwidth and low energy wireless transceiver using OFDM modulation scheme for mm-wave frequencies. It is resilient against channel fading and multipath propagation effects, prevalent in intra chip environments and provides high spectral efficiency irrespective of the antenna used. Analysis of the proposed transceiver in presence of these channel effects shows that it provides data rate of 195.32Gbps and per bit energy efficiency of 0.132pJ/bit with a low area overhead of 0.092mm^2 . A robust utilization aware channel access mechanism is integrated with single channel wireless networks to provide low overhead medium access control and improve channel utilization. We further enhance the OFDM based transceiver to enable simultaneous wireless transmissions by grouping different sub-channels of OFDM scheme into different bands and allocating them to individual transceivers in the wireless network.

Chapter 6

Conclusions and Future Directions

In this work, we have focused on developing wireless enabled interconnection networks for CPU/GPU heterogeneous architectures to address the challenges of architectural diversity and shared resource contention on communication performance. The major contributions of this thesis are:

1. HyWin, a novel interconnection with on-chip wireless links and sandboxed sub-networks is proposed in Chapter 2. It decouples different architectural components of heterogeneous architecture from a system wide network through customized, sandboxed sub-networks, which are interconnected at the global level through a wireless network. The custom sub-networks allow optimal network design for each subsystem to suit their respective communication needs. Besides improving performance, it significantly improves the flexibility and ease of design of interconnection topologies in heterogeneous networks.
2. Chapter 3 extends the concept to multi-chip CPU/GPU systems through a hybrid wireless interconnection network for connecting multiple CPU, GPU and memory chips. It implements an omnidirectional wireless link for remote communication between CPU and all other chips, while directional point-to-point wireless links provide sustained, high between communication between GPU and memory. An asynchronous TDMA control with collision detection is used to provide medium access control to different wireless transceivers in the omnidirectional network.
3. In Chapter 4, leveraging the hybrid wireless interconnection with sub-network topologies, WiSH protocol is proposed to provide scalable, low overhead and efficient cache coherence for many core architectures. The primary novelty of WiSH coherence is SpyDir protocol, which is a hierarchical combination of local directory protocol with global snoop coherence protocol. To the best of our knowledge, SpyDir is the first such protocol to implement snoop over directory coherence as opposed to existing implementations

of directory over snoopy protocols. Coherence performance in WiSH is further enhanced through share aware cache segmentation that exploits memory access patterns across the cores.

4. Finally, in Chapter 5, we design high bandwidth and low energy wireless links for the wireless network, that form the backbone for interconnection and coherence protocols proposed in our work. Towards this end, we have first analysed the on-chip wireless channel to understand wireless propagation in intra-chip environments and derive channel characteristics like path loss, dispersion and propagation delay. Utilizing this understanding and derived channel metrics, we design OFDM based wireless links that are robust against adverse channel effects and provide high spectral density at low per bit energy consumption.

Some of the possible future research directions to extend our work are:

- A seamless intra/inter chip wireless interconnection fully integrating intra-chip HyWin topology with chip-to-chip wireless network to provide uniform communication across heterogeneous compute subsystems both within and across multiple chips. Some of the major challenges for an intra/inter chip wireless network is interference between intra and inter chip communication, potential congestion in the wireless network and physical level implementations for wireless transmission across the chips.
- Wireless transceivers are mainly comprised of analog components, operating at very high frequencies. Integration of high speed analog components with digital circuits increase circuit design complexity and also impact their reliability. With wireless transceiver acting as gateway for communication in proposed interconnections, communication reliability is compromised in case of transceiver failures. Hence, a through analysis of communication robustness in wireless network and alternative communication paths in case of failures is a potential research direction for providing reliable and high performance communication.
- Extension of WiSH protocol to provide cross-architecture coherence in any heterogeneous systems offers another future direction.
- Although the communication challenges discussed in this work apply to most heterogeneous architectures and HyWin is easily extensible to them, it cannot be applied directly in its original state. Hence, a more generic framework of HyWin correlating abstract feature of communication paradigms in a heterogeneous system to the topological implementations of HyWin would further ease the integration of HyWin for any heterogeneous system.

Bibliography

- [1] R. H. Dennard, F. H. Gaensslen, H. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, “Design of ion-implanted mosfet’s with very small physical dimensions,” *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, Oct 1974.
- [2] M. Zapater and D. Atienza. (2019) Energy-efficient heterogeneous computing and memory architectures for latency-critical applications. [Online]. Available: <https://www.epfl.ch/labs/esl/research/thermal-aware/heterogeneous/>
- [3] T. Adegbiya, A. Rogacs, C. Patel, and A. Gordon-Ross, “Microprocessor optimizations for the internet of things: A survey,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 7–20, Jan 2018.
- [4] S. Exzellenz. (2017) Simulation on reconfigurable heterogeneous computer architectures. [Online]. Available: <https://rax5.informatik.uni-stuttgart.de/abteilungen/rechnerarchitektur/projekte/heterogeneous-computing.html>
- [5] AMD, “Compute cores and heterogeneous system architecture,” AMD, Tech. Rep., 2014. [Online]. Available: http://www.amd.com/Documents/Compute_Cores_Whitepaper.pdf
- [6] ——. (2019) Amd ryzenTM 3 pro 3200g processor. [Online]. Available: <https://www.amd.com/en/products/apu/amd-ryzen-3-pro-3200g>
- [7] ——. (2019) Amd ryzenTM 5 pro 3400g processor. [Online]. Available: <https://www.amd.com/en/products/apu/amd-ryzen-5-pro-3400g>
- [8] D. Kanter. (2011) Intel’s sandy bridge graphics architecture. [Online]. Available: <http://www.realworldtech.com/sandy-bridge-gpu/>
- [9] ——. (2012) Intel’s ivy bridge graphics architecture. [Online]. Available: <http://www.realworldtech.com/ivy-bridge-gpu/>
- [10] N. Kurd, M. Chowdhury, E. Burton, T. P. Thomas, C. Mozak, B. Boswell, M. Lal, A. Deval, J. Douglas, M. Elassal, A. Nalamalpu, T. M. Wilson, M. Merten, S. Chennupaty, W. Gomes, and R. Kumar, “5.9 haswell: A family of ia 22nm processors,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb 2014, pp. 112–113.
- [11] Intel. (2019) Intel coreTM i7-1065g7 processor. [Online]. Available: <https://ark.intel.com/content/www/us/en/ark/products/196597/intel-core-i7-1065g7-processor-8m-cache-up-to-3-90-ghz.html>
- [12] nVIDIA, *NVIDIA DGX-1 System Architecture-The Fastest Platform for Deep Learning*, nVIDIA, 2017. [Online]. Available: <http://www.nvidia.com/dgx1>

- [13] O. Kayiran, N. C. Nachiappan, A. Jog, R. Ausavarungnirun, M. T. Kandemir, G. H. Loh, O. Mutlu, and C. R. Das, "Managing gpu concurrency in heterogeneous architectures," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, Dec 2014, pp. 114–126.
- [14] A. K. Mishra, N. Vijaykrishnan, and C. R. Das, "A case for heterogeneous on-chip interconnects for cmps," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, June 2011, pp. 389–399.
- [15] C. Neeb and N. Wehn, "Designing efficient irregular networks for heterogeneous systems-on-chip," in *Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on*, 2006, pp. 665–672.
- [16] J. Yin, P. Zhou, S. S. Sapatnekar, and A. Zhai, "Energy-efficient time-division multiplexed hybrid-switched noc for heterogeneous multicore systems," in *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*, May 2014, pp. 293–303.
- [17] J. Wu, D. Dong, X. Liao, and L. Wang, "Chameleon: Adaptive energy-efficient heterogeneous network-on-chip," in *2015 33rd IEEE International Conference on Computer Design (ICCD)*, Oct 2015, pp. 419–422.
- [18] A. Flores, J. L. Aragon, and M. E. Acacio, "Heterogeneous interconnects for energy-efficient message management in cmps," *IEEE Transactions on Computers*, vol. 59, no. 1, pp. 16–28, Jan 2010.
- [19] J. Lee, S. Li, H. Kim, and S. Yalamanchili, "Adaptive virtual channel partitioning for network-on-chip in heterogeneous architectures," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 18, no. 4, pp. 48:1–48:28, Oct. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2504906>
- [20] Z. Li, N. Goswami, and T. Li, "iconn: A communication infrastructure for heterogeneous computing architectures," *J. Emerg. Technol. Comput. Syst.*, vol. 11, no. 4, pp. 42:1–42:23, Apr. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2700238>
- [21] J. Lee, S. Li, H. Kim, and S. Yalamanchili, "Design space exploration of on-chip ring interconnection for a cpu-gpu heterogeneous architecture," *J. Parallel Distrib. Comput.*, vol. 73, no. 12, pp. 1525–1538, Dec. 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.jpdc.2013.07.014>
- [22] R. Garg, G. Mallarapu, and S. P. Khatri, "A single-supply true voltage level shifter," in *2008 Design, Automation and Test in Europe*, March 2008, pp. 979–984.
- [23] N. Kadayinti, M. S. Baghini, and D. K. Sharma, "A clock synchronizer for repeaterless low swing on-chip links," *CoRR*, vol. abs/1510.04241, 2015. [Online]. Available: <http://arxiv.org/abs/1510.04241>
- [24] A. K. Mishra, O. Mutlu, and C. R. Das, "A heterogeneous multiple network-on-chip design: An application-aware approach," in *Design Automation Conference (DAC), 2013 50th ACM/EDAC/IEEE*, May 2013, pp. 1–10.
- [25] A. Shah, N. Mansoor, B. Johnstone, A. Ganguly, and S. L. Alarcon, "Heterogeneous photonic network-on-chip with dynamic bandwidth allocation," in *2014 27th IEEE International System-on-Chip Conference (SOCC)*, Sept 2014, pp. 249–254.
- [26] J. Yin, P. Zhou, A. Holey, S. S. Sapatnekar, and A. Zhai, "Energy-efficient non-minimal path on-chip interconnection network for heterogeneous systems," in *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '12. New York, NY, USA: ACM, 2012, pp. 57–62. [Online]. Available: <http://doi.acm.org/10.1145/2333660.2333675>

- [27] J. Fang, Z.-Y. Leng, S.-T. Liu, Z.-C. Yao, and X.-F. Sui, "Exploring heterogeneous noc design space in heterogeneous gpu-cpu architectures," *Journal of Computer Science and Technology*, vol. 30, no. 1, pp. 74–83, 2015. [Online]. Available: <http://dx.doi.org/10.1007/s11390-015-1505-6>
- [28] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, "Kilo-noc: A heterogeneous network-on-chip architecture for scalability and service guarantees," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, June 2011, pp. 401–412.
- [29] M. Lodde, J. Flich, and M. E. Acacio, "Heterogeneous noc design for efficient broadcast-based coherence protocol support," in *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on*, May 2012, pp. 59–66.
- [30] S. Deb, K. P. Chang, X. Yu, S. P. Sah, M. Cosic, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Design of an energy-efficient cmos-compatible noc architecture with millimeter-wave wireless interconnects," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2382–2396, Dec 2013.
- [31] C. Wang, W. H. Hu, and N. Bagherzadeh, "A wireless network-on-chip design for multicore platforms," in *2011 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing*, Feb 2011, pp. 409–416.
- [32] W. H. Hu, C. Wang, and N. Bagherzadeh, "Design and analysis of a mesh-based wireless network-on-chip," in *2012 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, Feb 2012, pp. 483–490.
- [33] A. Rezaei, F. Safaei, M. Daneshtalab, and H. Tenhunen, "Hiwa: A hierarchical wireless network-on-chip architecture," in *High Performance Computing Simulation (HPCS), 2014 International Conference on*, July 2014, pp. 499–505.
- [34] Intel. (2016) Intel hyper-threading technology. [Online]. Available: <http://www.intel.com/content/www/us/en/architecture-and-technology/hyper-threading/hyper-threading-technology.html>
- [35] F. Hamzaoglu, U. Arslan, N. Bisnik, S. Ghosh, M. B. Lal, N. Lindert, M. Meterelliyoz, R. B. Osborne, J. Park, S. Tomishima, Y. Wang, and K. Zhang, "A 1 gb 2 ghz 128 gb/s bandwidth embedded dram in 22 nm tri-gate cmos technology," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 150–157, Jan 2015.
- [36] R. Ubal, B. Jang, P. Mistry, D. Schaa, and D. Kaeli, "Multi2sim: A simulation framework for cpu-gpu computing," in *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '12. New York, NY, USA: ACM, 2012, pp. 335–344. [Online]. Available: <http://doi.acm.org/10.1145/2370816.2370865>
- [37] C. D. Spradling, "SPEC CPU2006 Benchmark Tools," *SIGARCH Computer Architecture News*, vol. 35, March 2007.
- [38] "AMD Accelerated Parallel Processing (APP) Software Development Kit (SDK) ," 2016. [Online]. Available: <http://developer.amd.com/sdks/amdappsdk/>
- [39] C. Lee, M. Potkonjak, and W. H. Mangione-Smith, "Mediabench: a tool for evaluating and synthesizing multimedia and communications systems," in *Proceedings of 30th Annual International Symposium on Microarchitecture*, Dec 1997, pp. 330–335.
- [40] U. Y. Ogras and R. Marculescu, "'it's a small world after all': Noc performance optimization via long-range link insertion," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 7, pp. 693–706, July 2006.

- [41] N. Agarwal, D. Nellans, E. Ebrahimi, T. F. Wenisch, J. Danskin, and S. W. Keckler, "Selective gpu caches to eliminate cpu-gpu hw cache coherence," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, March 2016, pp. 494–506.
- [42] H. K. Mondal, S. H. Gade, R. Kishore, and S. Deb, "Adaptive multi-voltage scaling in wireless noc for high performance low power applications," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 1315–1320.
- [43] H. K. Mondal, S. H. Gade, R. Kishore, S. Kaushik, and S. Deb, "Power efficient router architecture for wireless network-on-chip," in *2016 17th International Symposium on Quality Electronic Design (ISQED)*, March 2016, pp. 227–233.
- [44] A. Samaiyar, S. S. Ram, and S. Deb, "Millimeter-wave planar log periodic antenna for on-chip wireless interconnects," in *Antennas and Propagation (EuCAP), 2014 8th European Conference on*, April 2014, pp. 1007–1009.
- [45] D. Mayhew and V. Krishnan, "Pci express and advanced switching: evolutionary path to building next generation interconnects," in *11th Symposium on High Performance Interconnects, 2003. Proceedings.*, Aug 2003, pp. 21–29.
- [46] M. Harris. (2013, November) Unified memory in cuda 6. nVIDIA. [Online]. Available: <https://devblogs.nvidia.com/unified-memory-in-cuda-6/>
- [47] L. Barnes, "Multi-gpu programming," in *GPU Technology Conference*. nVIDIA, March 2013.
- [48] G. Kim, M. Lee, J. Jeong, and J. Kim, "Multi-gpu system design with memory networks," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, Dec 2014, pp. 484–495.
- [49] M. S. Birrittella, M. Debbage, R. Huggahalli, J. Kunz, T. Lovett, T. Rimmer, K. D. Underwood, and R. C. Zak, "Enabling scalable high-performance systems with the intel omni-path architecture," *IEEE Micro*, vol. 36, no. 4, pp. 38–47, July 2016.
- [50] I. Cutress. (2017, June) Amd's future in servers: New 7000 - series cpus launched and epyc analysis. Anandtech. [Online]. Available: <https://www.anandtech.com/show/11551/amds-future-in-servers-new-7000-series-cpus-launched-and-epyc-analysis/2>
- [51] NVIDIA. (2018) Nvidia nvswitch-the world's highest bandwidth on-node switch. NVIDIA. [Online]. Available: <http://images.nvidia.com/content/pdf/nvswitch-technical-overview.pdf>
- [52] S. Deb, K. Chang, X. Yu, S. P. Sah, M. Cosic, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Design of an energy-efficient cmos-compatible noc architecture with millimeter-wave wireless interconnects," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2382–2396, Dec 2013.
- [53] D. DiTomaso, A. Kodi, S. Kaya, and D. Matolak, "iwise: Inter-router wireless scalable express channels for network-on-chips (nocs) architecture," in *2011 IEEE 19th Annual Symposium on High Performance Interconnects*, Aug 2011, pp. 11–18.
- [54] X. Zhang, J. K. Lin, S. Wickramanayaka, S. Zhang, R. Weerasekera, R. Dutta, K. F. Chang, K.-J. Chui, H. Y. Li, D. S. Wee Ho, L. Ding, G. Katti, S. Bhattacharya, and D.-L. Kwong, "Heterogeneous 2.5d integration on through silicon interposer," *Applied Physics Reviews*, vol. 2, no. 2, p. 021308, 2015. [Online]. Available: <https://doi.org/10.1063/1.4921463>
- [55] J. H. Lau, "Evolution, challenge, and outlook of tsv, 3d ic integration and 3d silicon integration," in *2011 International Symposium on Advanced Packaging Materials (APM)*, Oct 2011, pp. 462–488.

- [56] G. H. Loh, "3d-stacked memory architectures for multi-core processors," in *Proceedings of the 35th Annual International Symposium on Computer Architecture*, ser. ISCA '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 453–464. [Online]. Available: <https://doi.org/10.1109/ISCA.2008.15>
- [57] L. Li, P. Ton, M. Nagar, and P. Chia, "Reliability challenges in 2.5d and 3d ic integration," in *2017 IEEE 67th Electronic Components and Technology Conference (ECTC)*, May 2017, pp. 1504–1509.
- [58] Intel, "Intel omni-path 4.8 tbps switch asic and platform," in *2016 IEEE Hot Chips 28 Symposium (HCS)*, Aug 2016, pp. 1–17.
- [59] M. S. Birrittella, M. Debbage, R. Huggahalli, J. Kunz, T. Lovett, T. Rimmer, K. D. Underwood, and R. C. Zak, "Intel® omni-path architecture: Enabling scalable, high performance fabrics," in *2015 IEEE 23rd Annual Symposium on High-Performance Interconnects*, Aug 2015, pp. 1–9.
- [60] WikiChip. (2018) Infinity fabric (if) - amd. [Online]. Available: https://en.wikichip.org/wiki/amd/infinity_fabric
- [61] R. Mahajan, R. Sankman, N. Patel, D. Kim, K. Aygun, Z. Qian, Y. Mekonnen, I. Salama, S. Sharan, D. Iyengar, and D. Mallik, "Embedded multi-die interconnect bridge (emib) – a high density, high bandwidth packaging interconnect," in *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*, May 2016, pp. 557–565.
- [62] M. Deo, "Enabling next-generation platforms using intel's 3d system-in-package technology," Intel, Tech. Rep., 2017.
- [63] X. Wu, Y. Ye, J. Xu, W. Zhang, W. Liu, M. Nikdast, and X. Wang, "Union: A unified inter/intrachip optical network for chip multiprocessors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 5, pp. 1082–1095, May 2014.
- [64] H. Matsutani, M. Koibuchi, I. Fujiwara, T. Kagami, Y. Take, T. Kuroda, P. Bogdan, R. Marculescu, and H. Amano, "Low-latency wireless 3d nocs via randomized shortcut chips," in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2014, pp. 1–6.
- [65] M. S. Shamim, N. Mansoor, R. S. Narde, V. Kothandapani, A. Ganguly, and J. Venkataraman, "A wireless interconnection framework for seamless inter and intra-chip communication in multichip systems," *IEEE Transactions on Computers*, vol. 66, no. 3, pp. 389–402, March 2017.
- [66] M. S. Shamim, M. M. Ahmed, N. Mansoor, and A. Ganguly, "Energy-efficient wireless interconnection framework for multichip systems with in-package memory stacks," in *2017 30th IEEE International System-on-Chip Conference (SOCC)*, Sept 2017, pp. 357–362.
- [67] S. Laha, S. Kaya, D. W. Matolak, W. Rayess, D. DiTomaso, and A. Kodi, "A new frontier in ultralow power wireless links: Network-on-chip and chip-to-chip interconnects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 186–198, Feb 2015.
- [68] A. Karkar, T. Mak, K. F. Tong, and A. Yakovlev, "A survey of emerging interconnects for on-chip efficient multicast and broadcast in many-cores," *IEEE Circuits and Systems Magazine*, vol. 16, no. 1, pp. 58–72, Firstquarter 2016.
- [69] S. H. Gade and S. Deb, "Hywin: Hybrid wireless noc with sandboxed sub-networks for cpu/gpu architectures," *IEEE Transactions on Computers*, vol. 66, no. 7, pp. 1145–1158, July 2017.
- [70] A. Kumar and M. Trivedi, "Intel xeon scalable processor architecture deep dive," June 2017. [Online]. Available: <http://www.primeline-solutions.de/files/intel-xeon-scalable-architecture-deep-dive1.pdf>

- [71] J. Lawley, “Understanding performance of pci express systems,” Xilinx, Whitepaper, 2014.
- [72] J. Nickolls and W. J. Dally, “The gpu computing era,” *IEEE Micro*, vol. 30, no. 2, pp. 56–69, March 2010.
- [73] A. Samaiyar, S. S. Ram, and S. Deb, “Millimeter-wave planar log periodic antenna for on-chip wireless interconnects,” in *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, April 2014, pp. 1007–1009.
- [74] X. Yu, J. Baylon, P. Wettin, D. Heo, P. P. Pande, and S. Mirabbasi, “Architecture and design of multi-channel millimeter-wave wireless noc,” *IEEE Design Test*, vol. 31, no. 6, pp. 19–28, Dec 2014.
- [75] B.-J. Kwak, N.-O. Song, and L. E. Miller, “Performance analysis of exponential backoff,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 343–355, April 2005.
- [76] C. Bienia and K. Li, “PARSEC 2.0: A New Benchmark Suite for Chip-Multiprocessors,” in *Proc. of the 5th Annual Workshop on Modeling, Benchmarking and Simulation*, June 2009.
- [77] S. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, “The SPLASH-2 Programs: Characterization and Methodological Considerations,” in *Proc. of the 22nd International Symposium on Computer Architecture*, June 1995.
- [78] “Amd accelerated parallel processing (app) software development kit (sdk).” [Online]. Available: <http://developer.amd.com/sdks/amdappsdk/>
- [79] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, S. H. Lee, and K. Skadron, “Rodinia: A benchmark suite for heterogeneous computing,” in *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, Oct 2009, pp. 44–54.
- [80] S. H. Gade, M. Sinha, S. S. Rout, and S. Deb, “Enabling reliable high throughput on-chip wireless communication for many core architectures,” in *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, July 2018, pp. 591–596.
- [81] S. H. Gade, S. S. Rout, M. Sinha, H. K. Mondal, W. Singh, and S. Deb, “A utilization aware robust channel access mechanism for wireless nocs,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2018, pp. 1–5.
- [82] H. K. Mondal, S. H. Gade, S. Kaushik, and S. Deb, “Adaptive multi-voltage scaling with utilization prediction for energy-efficient wireless noc,” *IEEE Transactions on Sustainable Computing*, vol. 2, no. 4, pp. 382–395, Oct 2017.
- [83] H. K. Mondal, S. H. Gade, R. Kishore, and S. Deb, “P2noc: Power- and performance-aware noc architectures for sustainable computing,” *Sustainable Computing: Informatics and Systems*, vol. 16, pp. 25 – 37, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S221053791630230X>
- [84] M. M. K. Martin, M. D. Hill, and D. J. Sorin, “Why on-chip cache coherence is here to stay,” *Commun. ACM*, vol. 55, no. 7, pp. 78–89, Jul. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2209249.2209269>
- [85] K. Strauss, X. Shen, and J. Torrellas, “Uncorq: Unconstrained snoop request delivery in embedded-ring multiprocessors,” in *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, Dec 2007, pp. 327–342.

- [86] N. Agarwal, L. S. Peh, and N. K. Jha, "In-network snoop ordering (inso): Snoopy coherence on unordered interconnects," in *2009 IEEE 15th International Symposium on High Performance Computer Architecture*, Feb 2009, pp. 67–78.
- [87] J. Zebchuk, B. Falsafi, and A. Moshovos, "Multi-grain coherence directories," in *2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec 2013, pp. 359–370.
- [88] D. Sanchez and C. Kozyrakis, "Scd: A scalable coherence directory with flexible sharer set encoding," in *IEEE International Symposium on High-Performance Comp Architecture*, Feb 2012, pp. 1–12.
- [89] J. Zebchuk, M. K. Qureshi, V. Srinivasan, and A. Moshovos, "A tagless coherence directory," in *2009 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Dec 2009, pp. 423–434.
- [90] H. Zhao, A. Shriraman, S. Dwarkadas, and V. Srinivasan, "Spatl: Honey, i shrunk the coherence directory," in *Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on*, Oct 2011, pp. 33–44.
- [91] B. K. Daya, C. H. O. Chen, S. Subramanian, W. C. Kwon, S. Park, T. Krishna, J. Holt, A. P. Chandrakasan, and L. S. Peh, "Scorpio: A 36-core research chip demonstrating snoopy coherence on a scalable mesh noc with in-network ordering," in *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, June 2014, pp. 25–36.
- [92] G. Kurian, J. E. Miller, J. Psota, J. Eastep, J. Liu, J. Michel, L. C. Kimerling, and A. Agarwal, "Atac: A 1000-core cache-coherent processor with on-chip optical network," in *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '10. New York, NY, USA: ACM, 2010, pp. 477–488. [Online]. Available: <http://doi.acm.org/10.1145/1854273.1854332>
- [93] J. L. Hennessy and D. A. Patterson, *Computer Architecture, Fifth Edition: A Quantitative Approach*, 5th ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.
- [94] A. Asaduzzaman and K. K. Chidella, "A novel directory based hybrid cache coherence protocol for shared memory multiprocessors," in *2016 IEEE International Symposium on Phased Array Systems and Technology (PAST)*, Oct 2016, pp. 1–6.
- [95] H. Zhao, O. Jang, W. Ding, Y. Zhang, M. Kandemir, and M. J. Irwin, "A hybrid noc design for cache coherence optimization for chip multiprocessors," in *DAC Design Automation Conference 2012*, June 2012, pp. 834–842.
- [96] B. Cuesta, A. Ros, M. E. Gómez, A. Robles, and J. Duato, "Increasing the effectiveness of directory caches by deactivating coherence for private memory blocks," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, June 2011, pp. 93–103.
- [97] A. Ros, M. Davari, and S. Kaxiras, "Hierarchical private/shared classification: The key to simple and efficient coherence for clustered cache hierarchies," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2015, pp. 186–197.
- [98] Wikichip. (2019) Chiplet. [Online]. Available: <https://en.wikichip.org/wiki/chiplet>
- [99] N. Beck, S. White, M. Paraschou, and S. Naffziger, "'zeppelin': An soc for multichip architectures," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, Feb 2018, pp. 40–42.

- [100] D. Schor. (2018, August) Amd announces threadripper 2, chiplets aid core scaling. WikiChip. [Online]. Available: <https://fuse.wikichip.org/news/1569/amd-announces-threadripper-2-chiplets-aid-core-scaling/>
- [101] R. Mahajan, R. Sankman, N. Patel, D. Kim, K. Aygun, Z. Qian, Y. Mekonnen, I. Salama, S. Sharan, D. Iyengar, and D. Mallik, "Embedded multi-die interconnect bridge (emib) – a high density, high bandwidth packaging interconnect," in *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*, May 2016, pp. 557–565.
- [102] J. Hruska. (2018, December) Intel uses new foveros 3d chip-stacking to build core, atom on same silicon. ExtremeTech. [Online]. Available: <https://www.extremetech.com/computing/282137-intel-uses-new-foveros-3d-chip-stacking-technology-to-build-core-atom-on-the-same-silicon>
- [103] T. Vijayaraghavan, Y. Eckert, G. H. Loh, M. J. Schulte, M. Ignatowski, B. M. Beckmann, W. C. Brantley, J. L. Greathouse, W. Huang, A. Karunanithi, O. Kayiran, M. Meswani, I. Paul, M. Poremba, S. Raasch, S. K. Reinhardt, G. Sadowski, and V. Sridharan, "Design and analysis of an apu for exascale computing," in *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Feb 2017, pp. 85–96.
- [104] A. Arunkumar, E. Bolotin, B. Cho, U. Milic, E. Ebrahimi, O. Villa, A. Jaleel, C.-J. Wu, and D. Nellans, "Mcm-gpu: Multi-chip-module gpus for continued performance scalability," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, ser. ISCA '17. New York, NY, USA: ACM, 2017, pp. 320–332. [Online]. Available: <http://doi.acm.org/10.1145/3079856.3080231>
- [105] C. Walker. (2017) New intel core processor combines high-performance cpu with custom discrete graphics from amd to enable sleeker, thinner devices. [Online]. Available: <https://newsroom.intel.com/editorials/new-intel-core-processor-combine-high-performance-cpu-discrete-graphics-sleek-thin-devices/>
- [106] H. Zhao, A. Shriraman, and S. Dwarkadas, "Space: Sharing pattern-based directory coherence for multicore scalability," in *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '10. New York, NY, USA: ACM, 2010, pp. 135–146. [Online]. Available: <http://doi.acm.org/10.1145/1854273.1854294>
- [107] O. Markish, O. Katz, B. Sheinman, D. Corcos, and D. Elad, "On-chip millimeter wave antennas and transceivers," in *Proceedings of the 9th International Symposium on Networks-on-Chip*, ser. NOCS '15. New York, NY, USA: ACM, 2015, pp. 11:1–11:7. [Online]. Available: <http://doi.acm.org/10.1145/2786572.2789983>
- [108] M. M. K. Martin, M. D. Hill, and D. A. Wood, "Token coherence: decoupling performance and correctness," in *Computer Architecture, 2003. Proceedings. 30th Annual International Symposium on*, June 2003, pp. 182–193.
- [109] N. Agarwal, L.-S. Peh, and N. K. Jha, "In-network coherence filtering: Snoopy coherence without broadcasts," in *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 42. New York, NY, USA: ACM, 2009, pp. 232–243. [Online]. Available: <http://doi.acm.org/10.1145/1669112.1669143>
- [110] R. G. Dreslinski, T. Manville, K. Sewell, R. Das, N. Pinckney, S. Satpathy, D. Blaauw, D. Sylvester, and T. Mudge, "Xpoint cache: Scaling existing bus-based coherence protocols for 2d and 3d many-core systems," in *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '12. New York, NY, USA: ACM, 2012, pp. 75–86. [Online]. Available: <http://doi.acm.org/10.1145/2370816.2370829>

- [111] A. Ros, M. E. Acacio, and J. M. Garcia, "A direct coherence protocol for many-core chip multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 12, pp. 1779–1792, Dec 2010.
- [112] M. Alisafae, "Spatiotemporal coherence tracking," in *2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, Dec 2012, pp. 341–350.
- [113] A. García-Guirado, R. Fernández-Pascual, and J. M. García, "Icci: In-cache coherence information," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 995–1014, April 2015.
- [114] A. Ros and A. Jimborean, "A hybrid static-dynamic classification for dual-consistency cache coherence," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 11, pp. 3101–3115, Nov 2016.
- [115] A. Kayi and T. El-Ghazawi, "An adaptive cache coherence protocol for chip multiprocessors," in *Proceedings of the Second International Forum on Next-Generation Multicore/Manycore Technologies*, ser. IFMT '10. New York, NY, USA: ACM, 2010, pp. 4:1–4:10. [Online]. Available: <http://doi.acm.org/10.1145/1882453.1882458>
- [116] A. Kayi, O. Serres, and T. El-Ghazawi, "Adaptive cache coherence mechanisms with producer-consumer sharing optimization for chip multiprocessors," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 316–328, Feb 2015.
- [117] P. Caheny, M. Casas, M. Moretó, H. Gloaguen, M. Saintes, E. Ayguadé, J. Labarta, and M. Valero, "Reducing cache coherence traffic with hierarchical directory cache and numa-aware runtime scheduling," in *Proceedings of the 2016 International Conference on Parallel Architectures and Compilation*, ser. PACT '16. New York, NY, USA: ACM, 2016, pp. 275–286. [Online]. Available: <http://doi.acm.org/10.1145/2967938.2967962>
- [118] B. K. Daya, L. S. Peh, and A. P. Chandrakasan, "Low-power on-chip network providing guaranteed services for snoopy coherent and artificial neural network systems," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2017, pp. 1–6.
- [119] S. Volos, C. Seiculescu, B. Grot, N. K. Pour, B. Falsafi, and G. D. Micheli, "Ccnoc: Specializing on-chip interconnects for energy efficiency in cache-coherent servers," in *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on*, May 2012, pp. 67–74.
- [120] T. Shreedhar and S. Deb, "Hierarchical cluster based noc design using wireless interconnects for coherence support," in *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, Jan 2016, pp. 63–68.
- [121] S. Deb, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Wireless noc as interconnection backbone for multicore chips: Promises and challenges," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 228–239, June 2012.
- [122] D. DiTomaso, A. Kodi, S. Kaya, and D. Matolak, "iwise: Inter-router wireless scalable express channels for network-on-chips (nocs) architecture," in *High Performance Interconnects (HOTI), 2011 IEEE 19th Annual Symposium on*, Aug 2011, pp. 11–18.
- [123] H. M. Cheema and A. Shamim, "The last barrier: on-chip antennas," *IEEE Microwave Magazine*, vol. 14, no. 1, pp. 79–91, Jan 2013.
- [124] K. Kim and K. K. O, "Characteristics of integrated dipole antennas on bulk, soi, and sos substrates for wireless communication," in *Proceedings of the IEEE 1998 International Interconnect Technology Conference (Cat. No.98EX102)*, June 1998, pp. 21–23.

- [125] T. Kikkawa, A. B. M. H. Rashid, and S. Watanabe, "Effect of silicon substrate on the transmission characteristics of integrated antenna," in *2003 IEEE Topical Conference on Wireless Communication Technology*, Oct 2003, pp. 144–145.
- [126] X. Guo, R. Li, and K. K. O, "Design guidelines for reducing the impact of metal interference structures on the performance on-chip antennas," in *IEEE Antennas and Propagation Society International Symposium. Digest. Held in conjunction with: USNC/CNC/URSI North American Radio Sci. Meeting (Cat. No.03CH37450)*, vol. 1, June 2003, pp. 606–609 vol.1.
- [127] E. Seok and K. O. Kenneth, "Design rules for improving predictability of on-chip antenna characteristics in the presence of other metal structures," in *Proceedings of the IEEE 2005 International Interconnect Technology Conference, 2005.*, June 2005, pp. 120–122.
- [128] A. Babakhani, X. Guan, A. Komijani, A. Natarajan, and A. Hajimiri, "A 77-ghz phased-array transceiver with on-chip antennas in silicon: Receiver and antennas," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2795–2806, Dec 2006.
- [129] D. Neculoiu, A. Muller, K. Tang, E. Laskin, and S. P. Voinigescu, "160 ghz on-chip dipole antenna structure in silicon technology," in *2007 International Semiconductor Conference*, vol. 1, Oct 2007, pp. 245–248.
- [130] A. Samaiyar, S. S. Ram, and S. Deb, "Millimeter-wave planar log periodic antenna for on-chip wireless interconnects," in *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, April 2014, pp. 1007–1009.
- [131] Y. P. Zhang, L. H. Guo, and M. Sun, "High transmission gain inverted-f antenna on low-resistivity si for wireless interconnect," *IEEE Electron Device Letters*, vol. 27, no. 5, pp. 374–376, May 2006.
- [132] A. Natarajan, A. Komijani, X. Guan, A. Babakhani, and A. Hajimiri, "A 77-ghz phased-array transceiver with on-chip antennas in silicon: Transmitter and local lo-path phase shifting," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, pp. 2807–2819, Dec 2006.
- [133] Y. P. Zhang, Z. M. Chen, and M. Sun, "Propagation mechanisms of radio waves over intra-chip channels with integrated antennas: Frequency-domain measurements and time-domain analysis," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 10, pp. 2900–2906, Oct 2007.
- [134] L. Yan and G. W. Hanson, "Wave propagation mechanisms for intra-chip communications," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 9, pp. 2715–2724, Sept 2009.
- [135] J. L. Volakis, *Antenna Engineering Handbook*, J. L. Volakis, Ed. McGraw-Hill Education, 2018.
- [136] P. Packan, S. Akbar, M. Armstrong, D. Bergstrom, M. Brazier, H. Deshpande, K. Dev, G. Ding, T. Ghani, O. Golonzka, W. Han, J. He, R. Heussner, R. James, J. Jopling, C. Kenyon, S.-H. Lee, M. Liu, S. Lodha, B. Mattis, A. Murthy, L. Neiberg, J. Neirynck, S. Pae, C. Parker, L. Pipes, J. Sebastian, J. Seiple, B. Sell, A. Sharma, S. Sivakumar, B. Song, A. St.Amour, K. Tone, T. Troeger, C. Weber, K. Zhang, Y. Luo, and S. Natarajan, "High performance 32nm logic technology featuring 2nd generation high-k + metal gate transistors," in *Electron Devices Meeting (IEDM), 2009 IEEE International*, Dec 2009, pp. 1–4.
- [137] J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.*, vol. 114, no. 2, p. 185–200, Oct. 1994.
- [138] D. Neculoiu, A. Muller, K. Tang, E. Laskin, and S. P. Voinigescu, "160 ghz on-chip dipole antenna structure in silicon technology," in *2007 International Semiconductor Conference*, vol. 1, Oct 2007, pp. 245–248.

- [139] CST. (2014) Cst microwave studio, transient solver. [Online]. Available: <https://www.cst.com/Products/CSTmws/TransientSolver>
- [140] S. H. Gade and S. Deb, "Achievable performance enhancements with mm-wave wireless interconnects in noc," in *Proceedings of the 9th International Symposium on Networks-on-Chip*, ser. NOCS '15. New York, NY, USA: ACM, 2015, pp. 29:1–29:2. [Online]. Available: <http://doi.acm.org/10.1145/2786572.2786584>
- [141] C. W. Byeon, C. H. Yoon, and C. S. Park, "A 67-mw 10.7-gb/s 60-ghz ook cmos transceiver for short-range wireless communications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 9, pp. 3391–3401, Sept 2013.
- [142] X. Yu, S. P. Sah, H. Rashtian, S. Mirabbasi, P. P. Pande, and D. Heo, "A 1.2-pj/bit 16-gb/s 60-ghz ook transmitter in 65-nm cmos for wireless network-on-chip," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 10, pp. 2357–2369, Oct 2014.
- [143] X. Yu, H. Rashtian, S. Mirabbasi, P. P. Pande, and D. Heo, "An 18.7-gb/s 60-ghz ook demodulator in 65-nm cmos for wireless network-on-chip," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 3, pp. 799–806, March 2015.
- [144] X. Yu, J. Baylon, P. Wettin, D. Heo, P. P. Pande, and S. Mirabbasi, "Architecture and design of multi-channel millimeter-wave wireless noc," *IEEE Design Test*, vol. 31, no. 6, pp. 19–28, Dec 2014.
- [145] V. Vijayakumaran, M. P. Yuvaraj, N. Mansoor, N. Nerurkar, A. Ganguly, and A. Kwasinski, "Cdma enabled wireless network-on-chip," *J. Emerg. Technol. Comput. Syst.*, vol. 10, no. 4, pp. 28:1–28:20, Jun. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2536778>
- [146] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.
- [147] AMD, "AMD Accelerated Parallel Processing (APP) Software Development Kit (SDK) ," 2016. [Online]. Available: <http://developer.amd.com/sdks/amdappsdk/>
- [148] D. Zhao, Y. Wang, and H. Wu, "Dual-channel binary-countdown medium access control in wireless network-on-chip," in *Proceedings of the 2nd International Conference on Nano-Networks*, ser. Nano-Net '07. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.
- [149] A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer, and C. Teuscher, "Scalable hybrid wireless network-on-chip architectures for multicore systems," *IEEE Transactions on Computers*, vol. 60, no. 10, pp. 1485–1502, Oct 2011.
- [150] N. Mansoor, M. P. Yuvaraj, and A. Ganguly, "A robust medium access mechanism for millimeter-wave wireless network-on-chip architecture," in *2013 IEEE International SOC Conference*, Sep. 2013, pp. 362–367.
- [151] M. Palesi, M. Collotta, A. Mineo, and V. Catania, "An efficient radio access control mechanism for wireless network-on-chip architectures," *Journal of Low Power Electronics and Applications*, vol. 5, no. 2, pp. 38–56, 2015.
- [152] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Noxim: An open, extensible and cycle-accurate network on chip simulator," in *Application-specific Systems, Architectures and Processors (ASAP), 2015 IEEE 26th International Conference on*, July 2015, pp. 162–163.

- [153] X. Yu, S. P. Sah, H. Rashtian, S. Mirabbasi, P. P. Pande, and D. Heo, "A 1.2-pj/bit 16-gb/s 60-ghz ook transmitter in 65-nm cmos for wireless network-on-chip," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 10, pp. 2357–2369, Oct 2014.
- [154] X. Yu, H. Rashtian, S. Mirabbasi, P. P. Pande, and D. Heo, "An 18.7-gb/s 60-ghz ook demodulator in 65-nm cmos for wireless network-on-chip," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 3, pp. 799–806, March 2015.