

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD.,
Petitioner

v.

NETWORK SYSTEM TECHNOLOGIES, LLC,
Patent Owner

U.S. PATENT NO. 8,086,800

Case IPR2024-00339

DECLARATION OF TODD MOWRY, PH.D.

TABLE OF CONTENTS

I.	BACKGROUND AND QUALIFICATIONS.....	2
II.	MATERIALS AND OTHER INFORMATION CONSIDERED	5
III.	UNDERSTANDING OF PATENT LAW	6
A.	The Level of Ordinary Skill in the Art.....	6
B.	Obviousness	7
IV.	SUMMARY OF OPINIONS.....	10
V.	Overview of the Technology	10
A.	Systems-on-Chip (SoCs).....	10
B.	Networks-on-Chip (NoCs).....	11
C.	Network Optimization.....	13
VI.	THE '800 PATENT	15
A.	Claims	15
B.	Summary of the Specification.....	17
C.	Summary of Prosecution History	18
VII.	LEVEL OF ORDINARY SKILL IN THE ART	18
VIII.	OVERVIEW OF THE ASSERTED PRIOR ART	20
A.	Rădulescu	20
B.	Bahadıroglu	21
C.	Petersen	22
D.	Rădulescu-WO	23
IX.	SPECIFIC GROUNDS FOR PETITION	24

A. Ground 1: Claims 1, 2, 4, 6, 8–12, 13, and 18–21 Are Rendered Obvious by Rădulescu in view of Bahadiroglu	24
1. Claim 12	30
2. Claim 13: “The integrated circuit of claim 12, wherein the plurality of interconnect modules manages the communication connection freeing the plurality of processing modules to perform data processing.”	65
3. Claim 18: “The integrated circuit of claim 12, wherein the first processing module provides a first flushing command to flush the first memory and a second flushing command to flush the second memory so that the first data and the second data are sent without waiting to reach the first amount and the second amount.”	66
4. Claim 19: “The integrated circuit of claim 18, wherein the first processing module provides the first flushing command and the second flushing command at least one of independently and simultaneously.”	68
5. Claim 20: “The integrated circuit of claim 18, wherein the second flushing command includes an indication to flush the second memory so that the first interconnect module forwards the second flushing command to the second interconnect module for flushing the second data from the second memory.”	69
6. Claim 21: “The integrated circuit of claim 12, wherein the first processing module includes flush information in a header of a message to cause all messages from a same connection that have been sent earlier to be flushed from at least one of the first memory and the second memory.”	71
7. Claim 10	71
8. Claim 1	75
9. Claim 2: “The integrated circuit according to claim 1, wherein the first interface unit includes a first unbuffer-flag unit for providing a first unbuffer signal and the second interface unit includes a second unbuffer-flag unit for providing a second	

unbuffer signal, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to the first unbuffer signal and the second unbuffer signal, respectively, even if less than said first optimal amount and said second optimal amount of data is buffered in said first wrapper and said second wrapper.”	78
10. Claim 4: “The integrated circuit according to claim 2, wherein said first unbuffer signal is initiated by the slave and said second unbuffer signal is initiated by the master.”	79
11. Claim 6: “The integrated circuit according to claim 2, wherein the master is adapted to initiate said first unbuffer signal and said second unbuffer signal.”	80
12. Claim 8: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using messages of a message-passing communication scheme, wherein the messages contain a message header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said message header, respectively, even if less than said first amount and said second amount of data is buffered in said first wrapper and said second wrapper.”	80
13. Claim 9: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using packets of a packet based communication scheme, wherein the packets contain a packet header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said packet header, respectively, even if less than said first amount and said second amount of data is buffered in said first and second wrapper.”	81
14. Claim 11	82
B. Ground 2: Claims 1, 2, 4, 6, 8–12, 13, and 18–21 are Rendered Obvious by Rădulescu in view of Petersen.....	85
1. Claim 12	91

2. Claim 13: “The integrated circuit of claim 12, wherein the plurality of interconnect modules manages the communication connection freeing the plurality of processing modules to perform data processing.” 98
3. Claim 18: “The integrated circuit of claim 12, wherein the first processing module provides a first flushing command to flush the first memory and a second flushing command to flush the second memory so that the first data and the second data are sent without waiting to reach the first amount and the second amount.” 98
4. Claim 19: “The integrated circuit of claim 18, wherein the first processing module provides the first flushing command and the second flushing command at least one of independently and simultaneously.” 98
5. Claim 20: “The integrated circuit of claim 18, wherein the second flushing command includes an indication to flush the second memory so that the first interconnect module forwards the second flushing command to the second interconnect module for flushing the second data from the second memory.” 98
6. Claim 21: “The integrated circuit of claim 12, wherein the first processing module includes flush information in a header of a message to cause all messages from a same connection that have been sent earlier to be flushed from at least one of the first memory and the second memory.” 99
7. Claim 10 99
8. Claim 1 102
9. Claim 2: “The integrated circuit according to claim 1, wherein the first interface unit includes a first unbuffer-flag unit for providing a first unbuffer signal and the second interface unit includes a second unbuffer-flag unit for providing a second unbuffer signal, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to the first unbuffer signal and the second unbuffer signal, respectively, even if less than said first optimal amount and said

second optimal amount of data is buffered in said first wrapper and said second wrapper.”	104
10. Claim 4: “The integrated circuit according to claim 2, wherein said first unbuffer signal is initiated by the slave and said second unbuffer signal is initiated by the master.”	104
11. Claim 6: “The integrated circuit according to claim 2, wherein the master is adapted to initiate said first unbuffer signal and said second unbuffer signal.”	105
12. Claim 8: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using messages of a message-passing communication scheme, wherein the messages contain a message header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said message header, respectively, even if less than said first amount and said second amount of data is buffered in said first wrapper and said second wrapper.”	105
13. Claim 9: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using packets of a packet based communication scheme, wherein the packets contain a packet header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said packet header, respectively, even if less than said first amount and said second amount of data is buffered in said first and second wrapper.”	105
14. Claim 11	105
C. Ground 3: Claims 1, 2, 4, 6, 8–12, 13, and 18–21 are Rendered Obvious by Rădulescu in view of Bahadiroglu and Rădulescu-WO.....	108
1. Claim 12	110
2. Claim 13: “The integrated circuit of claim 12, wherein the plurality of interconnect modules manages the communication connection freeing the plurality of processing modules to perform data processing.”	115

3. Claim 18: “The integrated circuit of claim 12, wherein the first processing module provides a first flushing command to flush the first memory and a second flushing command to flush the second memory so that the first data and the second data are sent without waiting to reach the first amount and the second amount.” 115
4. Claim 19: “The integrated circuit of claim 18, wherein the first processing module provides the first flushing command and the second flushing command at least one of independently and simultaneously.” 116
5. Claim 20: “The integrated circuit of claim 18, wherein the second flushing command includes an indication to flush the second memory so that the first interconnect module forwards the second flushing command to the second interconnect module for flushing the second data from the second memory.” 116
6. Claim 21: “The integrated circuit of claim 12, wherein the first processing module includes flush information in a header of a message to cause all messages from a same connection that have been sent earlier to be flushed from at least one of the first memory and the second memory.” 116
7. Claim 10 116
8. Claim 1 119
9. Claim 2: “The integrated circuit according to claim 1, wherein the first interface unit includes a first unbuffer-flag unit for providing a first unbuffer signal and the second interface unit includes a second unbuffer-flag unit for providing a second unbuffer signal, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to the first unbuffer signal and the second unbuffer signal, respectively, even if less than said first optimal amount and said second optimal amount of data is buffered in said first wrapper and said second wrapper.” 122
10. Claim 4: “The integrated circuit according to claim 2, wherein said first unbuffer signal is initiated by the slave and said second unbuffer signal is initiated by the master.” 122

11. Claim 6: “The integrated circuit according to claim 2, wherein the master is adapted to initiate said first unbuffer signal and said second unbuffer signal.”	122
12. Claim 8: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using messages of a message-passing communication scheme, wherein the messages contain a message header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said message header, respectively, even if less than said first amount and said second amount of data is buffered in said first wrapper and said second wrapper.”	122
13. Claim 9	122
14. Claim 11	123
D. Ground 4: Claims 1, 2, 4, 6, 8–12, 13, and 18–21 are Rendered Obvious by Rădulescu in view of Petersen and Rădulescu-WO	125
1. Claim 12	125
2. Claim 13: “The integrated circuit of claim 12, wherein the plurality of interconnect modules manages the communication connection freeing the plurality of processing modules to perform data processing.”	127
3. Claim 18: “The integrated circuit of claim 12, wherein the first processing module provides a first flushing command to flush the first memory and a second flushing command to flush the second memory so that the first data and the second data are sent without waiting to reach the first amount and the second amount.”	128
4. Claim 19: “The integrated circuit of claim 18, wherein the first processing module provides the first flushing command and the second flushing command at least one of independently and simultaneously.”	128
5. Claim 20: “The integrated circuit of claim 18, wherein the second flushing command includes an indication to flush the	

- second memory so that the first interconnect module forwards the second flushing command to the second interconnect module for flushing the second data from the second memory.” 128
6. Claim 21: “The integrated circuit of claim 12, wherein the first processing module includes flush information in a header of a message to cause all messages from a same connection that have been sent earlier to be flushed from at least one of the first memory and the second memory.” 128
7. Claim 10 128
8. Claim 1 131
9. Claim 2: “The integrated circuit according to claim 1, wherein the first interface unit includes a first unbuffer-flag unit for providing a first unbuffer signal and the second interface unit includes a second unbuffer-flag unit for providing a second unbuffer signal, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to the first unbuffer signal and the second unbuffer signal, respectively, even if less than said first optimal amount and said second optimal amount of data is buffered in said first wrapper and said second wrapper.” 134
10. Claim 4: “The integrated circuit according to claim 2, wherein said first unbuffer signal is initiated by the slave and said second unbuffer signal is initiated by the master.” 134
11. Claim 6: “The integrated circuit according to claim 2, wherein the master is adapted to initiate said first unbuffer signal and said second unbuffer signal.” 134
12. Claim 8: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using messages of a message-passing communication scheme, wherein the messages contain a message header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said message header, respectively, even if less than said first amount and said second amount of data is buffered in said first wrapper and said second wrapper.” 134

13.	Claim 9: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using packets of a packet based communication scheme, wherein the packets contain a packet header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said packet header, respectively, even if less than said first amount and said second amount of data is buffered in said first and second wrapper.”.....	135
14.	Claim 11	135
X.	SECONDARY CONSIDERATIONS	137
XI.	CONCLUSION	138

LIST OF EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 8,086,800 (“the ’800 Patent”)
1002	Declaration of Todd Mowry in Support of Petition for <i>Inter Partes</i> Review of U.S. Patent No. 8,086,800
1003	Curriculum Vitae for Todd Mowry
1004	Prosecution History of U.S. Patent No. 8,086,800
1005	Andrei Rădulescu et al., <i>An Efficient On-Chip Network Interface Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Configuration</i> , 2 PROC. DESIGN, AUTOMATION AND TEST IN EUR. CONF. & EXHIBITION 878 (2004) (“Rădulescu”)
1006	U.S. Patent Application Publication No. 2002/0186660 to Murat I. Bahadiroglu et al. (“Bahadiroglu”)
1007	U.S. Patent No. 5,434,872 to Brian Petersen et al. (“Petersen”)
1008	International Patent Publication No. WO 2004/034176 A2 to Rădulescu et al. (“Rădulescu-WO”)
1009	Axel Jantsch and Hannu Tenhunen, <i>Networks on Chip</i> (2003)
1010	Kees Goossens et al., <i>Chapter 4: Guaranteeing the Quality of Services in Networks on Chip</i> , NETWORKS ON CHIP, at 61–82 (2003)
1011	Defs.’ Joint Opening Claim Construction Br., <i>Network Sys. Techs., LLC v. Qualcomm Inc.</i> , No. 1:22-cv-1331, Dkt. 77 (E.D. Tex. Oct. 12, 2023)
1012	Amit Goel & William R. Lee, <i>Formal Verification of an IBM CoreConnect™ Processor Local Bus Arbiter Core</i> , PROCEEDINGS OF THE 37TH ANNUAL DESIGN AUTOMATION CONFERENCE 196–200 (2000)
1013	Luca Benini & Giovanni De Micheli, <i>Networks on Chips: A New SoC Paradigm</i> , 35 COMPUTER 70–78 (2002)
1014	Erik Jan Marinissen, <i>Wrapper Design for Embedded Core Test</i> , IEEE INT’L TEST CONFERENCE (ITC) 1–10 (2000)
1015	Terry Tao Ye, et al., <i>Packetization and routing analysis of on-chip multiprocessor networks</i> , JOURNAL OF SYSTEMS

	ARCHITECTURE, Vol. 50, Issues 2-3, pp 81-104 (2004)
1016	Zhonghai Lu, <i>et al.</i> , Network-on-Chip Assembler Language, RETRIEVED FROM INSTITUTE OF MICROELECTRONICS AND INFORMATION TECHNOLOGY, ROYAL INSTITUTE OF TECHNOLOGY (KTH) (2003); available at https://people.kth.se/~zhonghai/Archive/NoC-AL-V01.pdf
1017	Declaration of Gordon MacPherson
1018	David E. Culler, <i>et al.</i> , <i>Parallel Computer Architecture</i> (1998)
1019	John Heinlein, <i>et al.</i> , <i>Coherent Block Data Transfer in the FLASH Multiprocessor</i> , IEEE Proceedings 11th International Parallel Processing Symposium, Genoa, Switzerland, pp. 18-27 (1997)
1020	Pl.’s Opening Claim Construction Br., <i>Network Sys. Techs., LLC v. Tex. Instruments Inc.</i> , No. 2:22-cv-00482-RWS, Dkt. 98 (E.D. Tex. Sept. 27, 2023)
1021	Pl.’s Responsive Claim Construction Br., <i>Network Sys. Techs., LLC v. Qualcomm Inc.</i> , No. 1:22-cv-1331-DAE, Dkt. 81 (W.D. Tex. Nov. 16, 2023)
1022	Wael Badawy, <i>System on Chip: The Challenge and Opportunities</i> , SYSTEM ON CHIP FOR REAL-TIME APPLICATIONS, 3–16 (2003)
1023	Ahmed Amine Jerraya & Wayne Wolf, <i>Multiprocessor Systems-on-Chips</i> (2005)
1024	Ann Marie Rincorn et al., <i>The Changing Landscape of System-on-a-Chip Design</i> , IEEE 1999 CUSTOM INTEGRATED CIRCUITS CONF. 83–90 (1999)
1025	Shashi Kumar <i>et al.</i> , <i>A Network on Chip Architecture and Design Methodology</i> , PROCEEDINGS OF THE IEEE COMPUTER SOCIETY ANNUAL SYMPOSIUM ON VLSI (ISVLSI’02) (2002)
1026	Luca Benini & Giovanni De Micheli, <i>Chapter 16: Energy-efficient system-level design</i> , POWER AWARE DESIGN METHODOLOGIES 473–516 (2002)
1027	Rudolf F. Graf, MODERN DICTIONARY OF ELECTRONICS (7th ed. 1999)
1028	David Howard, THE PENGUIN DICTIONARY OF ELECTRONICS (4th

	ed. 2005)
1029	Douglas A. Downing <i>et al.</i> , DICTIONARY OF COMPUTER AND INTERNET TERMS (8th ed. 2003)
1030	John Heinlein <i>et al.</i> , <i>Integration of Message Passing and Shared Memory in the Stanford FLASH Multiprocessor</i> , ACM SIGOPS OPERATING SYSTEMS REVIEW, Vol. 28, Issue 5, pp 38–50 (1994)
1031	Hugo De Man, <i>System-on-chip design: Impact on Education and Research</i> , IEEE DESIGN & TEST OF COMPUTERS, Vol. 16, No. 3, pp. 11-19, (July-Sept. 1999)
1032	William J. Dally, <i>et al.</i> , <i>Route packets, not wires: on-chip interconnection networks</i> , PROCEEDINGS OF THE 38TH DESIGN AUTOMATION CONFERENCE (IEEE Cat. No.01CH37232), Las Vegas, NV, USA, 2001, pp. 684-689 (2001)
1033	Andrei Rădulescu, <i>et al.</i> , <i>Communication Services For Networks On Chip</i> . SIGNAL PROCESSING AND COMMUNICATIONS (2000)
1034	Jose Duato, <i>et al.</i> , <i>Interconnection Networks: An Engineering Approach</i> , IEEE COMPUTER SOCIETY PRESS, Los Alamitos, CA (2002)
1035	Ahmed Hemani, <i>et al.</i> , <i>Network on Chip : An architecture for billion transistor era</i> (2001)
1036	Daniel Wiklund, <i>et al.</i> , <i>Design of a system-on-chip switched network and its design support</i> , IEEE 2002 INTERNATIONAL CONFERENCE ON COMMUNICATIONS, CIRCUITS AND SYSTEMS AND WEST SINO EXPOSITIONS, Chengdu, China, 2002, pp. 1279-1283 Vol.2

Declaration of Dr. Mowry for *Inter Partes* Review of U.S. Patent No. 8,086,800

I, Dr. Todd Mowry, declare as follows:

1. My name is Todd Mowry.
2. I have been retained as an expert witness on behalf of Samsung Electronics Co., Ltd., (“Samsung” or “Petitioner”). In this declaration, I provide my opinions as to whether certain claims of U.S. Patent No. 8,086,800 (the “’800 Patent”) (Ex. 1001) are valid in view of the prior art discussed herein. I also provide herein the technical bases for these opinions, as appropriate.
3. This declaration contains statements of my opinions formed to date, and the bases and rationale for these opinions. I may offer additional opinions based on further review of materials in this case, including opinions and/or testimony of other expert witnesses.
4. I am being compensated for my time in connection with this Petition at my standard consulting rate of \$650 per hour. My compensation is not affected by the outcome of this matter.
5. I have considered information from various sources in forming my opinions expressed herein, including the exhibits identified at the beginning of this declaration. I may review additional documents filed in connection with this proceeding as they become available.
6. I am not currently, and have not at any time in the past been, an employee of Samsung. I have no financial interest in Samsung.

I. BACKGROUND AND QUALIFICATIONS

7. I am a Professor in the Department of Computer Science at Carnegie Mellon University. I also have a courtesy appointment in the Department of Electrical and Computer Engineering. I have served on the faculty of Carnegie Mellon University for twenty-six (26) years starting in 1997 through the present (2023).

8. I also served on the faculty of the University of Toronto for four (4) years between 1993 and 1997, in the Department of Electrical and Computer Engineering and a courtesy appointment in the Department of Computer Science. Prior to that appointment, I served as a Graduate Research Assistant in the Department of Electrical Engineering at Stanford University for four (4) years between 1989 and 1993. As a faculty member, I have taught and continue to teach courses and directed research in computer architecture, parallel processing, computer systems and software, compiler optimization, and operating systems.

9. I received a B.S. degree in Electrical Engineering with Highest Distinction from the University of Virginia in May 1988. I received an M.S. in Electrical Engineering from Stanford University in June 1989, and a Ph.D. in Electrical Engineering from Stanford University in March 1994.

10. I have worked in the computer industry in various capacities. I was a part-time Computer Architect and then Computer Architecture Consultant at Silicon

Declaration of Dr. Mowry for *Inter Partes* Review of U.S. Patent No. 8,086,800

Graphics, Inc. in Mountain View, California (formerly MIPS Computer Systems in Sunnyvale, California) from 1989 to 1993 and 1993 to 1996, respectively. I was a Visiting Scientist at IBM in Toronto from 1996 to 2004. During that same time period (1996 to 2004), I was also a Member of the Technical Advisory Board of SandCraft, Inc. in Santa Clara, California. I was the Director of the Intel Research Pittsburgh Lab at Intel Corporation in Pittsburgh, Pennsylvania from 2004 to 2007.

11. I have authored 19 journal articles and over 60 conference papers. I am also an inventor on 5 patents.

12. I am the recipient of several honors and awards: the Arthur Samuel Thesis Award (awarded by the Stanford Computer Science department to the top two Ph.D. theses in a given year), several IBM Faculty Development Awards (1996, 1997, 1998, 2000, 2001, 2002, and 2003), several Best Paper Awards (the Second Symposium on Operating Systems Design and Implementation in 1996; the 20th International Conference on Data Engineering (ICDE) in 2004), the Alfred P. Sloan Research Fellowship (awarded to researchers in recognition of distinguished performance and a unique potential to make substantial contributions to their field), the Most Thought-Provoking Idea Award in 2004 (awarded by the Architectural Support for Programming Languages and Operating Systems (ASPLOS)), the TR100 Award in 1999 (awarded by MIT's Technology Review magazine to the top

100 most promising young innovators in science and technology), and I became an ACM Fellow in 2016.

13. I am a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association of Computing Machinery (ACM). I was the Editor-in-Chief of ACM Transactions on Computer Systems from 2013 through 2018, which is the premier journal for computer systems research. (I was an Associate Editor for the journal prior to that, since 2001.) I was the Program Chair of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) in 2010. I was the Co-Program Chair of the International Conference on Parallel Architectures and Compilation Techniques (PACT) in 2001. I have been on the programming committee in various years for ASPLOS, the International Symposium on Computer Architecture (ISCA), the International Symposium on Microarchitectures, and the Workshop on Architectural and System Support for Improving Format.

14. I have authored numerous publications relating to systems-on-chips (SOCs) comprised of multiple processors that communicate with each other via networks-on-chips (NOCs). For example, when the Association of Computing Machinery (ACM) made me an ACM Fellow in 2016, they explicitly recognized my research contributions in the area of thread-level speculation, which was a new technique specifically motivated by these architectures. As other examples, my

work in the Log-Based Architectures project proposed a novel use of SOC's where logs were captured and shipped across NOCs to other processors on the same chip, and my work on on-chip memory compression helped to reduce the load on the NOC (and also led to an interesting observation about how memory compression had an unintended consequence on NOC energy, which we were able to fix).

15. I am also the named inventor on multiple issued patents relating to processor architecture.

16. My qualifications and publications are set forth more fully in my curriculum vitae, attached as Ex. 1003.

II. MATERIALS AND OTHER INFORMATION CONSIDERED

17. In forming the opinions expressed in this Declaration, I relied upon my education and experience in the relevant field of the art and have considered the viewpoint of a person having ordinary skill in the art (POSITA) at the time of the alleged invention.

18. I have considered the materials referenced herein, including the '800 Patent (Ex. 1001), the file history of the '800 Patent (Ex. 1004), the Petition, and other documents listed in the Exhibit List of the Petition, including:

Description	Date of Public Availability
Andrei Rădulescu et al., <i>An Efficient On-Chip Network Interface Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Configuration</i> , 2 PROC. DESIGN,	February 20, 2004

AUTOMATION AND TEST IN EUR. CONF. & EXHIBITION 878 (2004) (“Rădulescu”) (Ex. 1005)	
U.S. Patent Application Publication No. 2002/0186660 to Bahadiroglu <i>et al.</i> (“Bahadiroglu”) (Ex. 1006)	December 12, 2002
U.S. Patent No. 5,434,872 to Brian Petersen <i>et al.</i> (“Petersen”) (Ex. 1007)	July 18, 1995
International Patent Publication No. WO 2004/034176 A2 to Rădulescu <i>et al.</i> (“Rădulescu-WO”) (Ex. 1008)	April 22, 2004

19. The references listed above are prior art to the '800 Patent.

20. Only one of the prior art references relied on in this Petition was cited or discussed during original prosecution. However, the arguments presented in this declaration are not the same or substantially the same as those considered during prosecution.

III. UNDERSTANDING OF PATENT LAW

21. I am not an attorney. For purposes of this declaration, I have been informed about certain aspects of the law that are relevant to my opinions. My understanding of the law is as listed below.

A. The Level of Ordinary Skill in the Art

22. It is my understanding that the level of ordinary skill in the art is based on a study of the patents at issue and their file histories, a study of the prior art cited therein, and knowledge of the following:

- The level of education and experience of persons actively working in the field at the time the subject matter at issue was developed;

- The types of problems encountered in the art at the time the subject matter was developed;
- The prior art patents and publications;
- The activities of others working in that field;
- Prior art solutions to those problems; and
- The sophistication of the technology at issue in this case.

23. It is also my understand that these factors are not exhaustive, and merely a useful guide to determine the level of ordinary skill in the art.

B. Obviousness

24. I have been informed and understand that a patent claim can be considered to have been obvious to a POSITA at the time the application was filed. This means that, even if all of the requirements of a claim are not found in a single prior art reference, the claim is not patentable if the differences between the subject matter in the prior art and the subject matter in the claim would have been obvious to a POSITA at the time the application was filed.

25. I have been informed and understand that a determination of whether a claim would have been obvious should be based upon several factors, including, among others:

- the level of ordinary skill in the art at the time the application was filed;
- the scope and content of the prior art; and

- what differences, if any, existed between the claimed invention and the prior art.

26. I have been informed and understand that the teachings of two or more references may be combined in the same way as disclosed in the claims, if such a combination would have been obvious to a POSITA. In determining whether a combination based on either a single reference or multiple references would have been obvious, it is appropriate to consider at least the following factors:

- whether the teachings of the prior art references disclose known concepts combined in familiar ways, which, when combined, would yield predictable results;
- whether a POSITA could implement a predictable variation, and would see the benefit of doing so;
- whether the claimed elements represent one of a limited number of known design choices, and would have a reasonable expectation of success by a POSITA;
- whether a POSITA would have recognized a reason to combine known elements in the manner described in the claims;
- whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; and
- whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way.

27. I understand that a POSITA has ordinary creativity, and is not an automaton.

28. I understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the patent being considered.

29. I understand that prior art to the '800 Patent includes patents and printed publications in the relevant art that predate the Priority Date of the '800 Patent.

30. I understand that certain factors—often called “secondary considerations”—may support or rebut an assertion of obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the alleged invention, skepticism of those having ordinary skill in the art at the time of the alleged invention, unexpected results of the alleged invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field.

31. I further understand that there must be a nexus—a connection—between any such secondary considerations and the alleged invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

IV. SUMMARY OF OPINIONS

32. It is my opinion that claims 1, 2, 4, 6, 8–12, 13, and 18–21 of the '800 Patent are rendered obvious based on the following combinations: (1) Rădulescu in view of Bahadiroglu; (2) Rădulescu in view of Petersen; (3) Rădulescu in view of and Bahadiroglu and Rădulescu-WO; and (4) Rădulescu in view of Petersen and Rădulescu-WO.

V. OVERVIEW OF THE TECHNOLOGY

A. Systems-on-Chip (SoCs)

33. Computer systems are comprised of a number of different components, including processors (e.g, CPUs, GPUs, etc.), memory (e.g., SRAM, DRAM, etc.) and various input/output (I/O) devices. As the capacity of integrated circuit (IC) technology has improved exponentially over the years, it has enabled larger portions of computer systems to be integrated onto single chips. For example, in the early 1970s, Intel was able to integrate an entire CPU onto a single chip: e.g., the Intel 4004, which was the first commercial “microprocessor”. Compared with CPUs during the 1960s (with their numerous separate chips on motherboards), single-chip microprocessors offered significant cost and energy savings.

34. As IC capacity has continued to increase well beyond what is needed to implement a single processor, it has enabled “Systems-on-Chip” (SoC) designs that include even more system components on a single chip: usually some combination of processors (often including multiple CPUs and possibly accelerators

such as GPUs), portions of the memory hierarchy, and possibly I/O devices or controllers. SoCs were well-known by the late 1990s, and are integrated on a silicon die. *See, e.g.*, Ex. 1009 (Jantsch) at 4 (native page vii) (“During the 1990s more and more processor cores and large reusable components have been integrated on a single silicon die, which has become known under the label System on Chip (SoC).”); Ex. 1023 (Wolf) at 5 (“An SoC is an integrated circuit that implements most or all of the functions of a complete electronic system.”); Ex. 1022 (Badawy) at 1 (“With the increase in the number of logic gates that can be implemented on a single chip, various functionalities ... can be integrated into a single silicon chip, realizing an entire system on chip (SoC)”). Thus, SoCs can also be referenced as System on Silicon. *See, e.g.*, Ex. 1022 (Badawy) at 1 (An SoC “is also known as a system on silicon”); Ex. 1031 (Man) at 12 (“An SOC thus encapsulates system knowledge in a single medium—silicon—which blurs the separation between component design and system design.”).

B. Networks-on-Chip (NoCs)

35. To enable the various system components within a chip for an SoC design to communicate with each other, an on-chip interconnect is needed. One such on-chip interconnect is often referred to as a “Network on Chip” (NoC). Rather than directly wiring together each pair of on-chip components via dedicated hardware FIFO buffers and wires, the idea of a NoC design is to create a more general on-chip

interconnect that routes packets from sources to destinations, often traversing multiple hops through on-chip routing switches. The rationale for this NoC approach was described in the influential paper by Bill Dally and Brian Towles that appeared in the Design Automation Conference (DAC)—the premier IC technology conference—in June, 2001. Ex. 1032 (Dally).

36. By integrating the interconnection network fully onto a chip, NoC designs can potentially provide lower cost, lower latency, and higher bandwidth relative to off-chip designs. To take full advantage of these potential benefits and to provide scalability as the number of on-chip components increases, NoCs have borrowed many design features from scalable interconnection networks that were designed for large-scale parallel machines in the 1990s. Ex. 1018 (Culler) at 17-96 (native pages 749-828); Ex. 1033 (Rădulescu II) at 277 (“Networks have been the subject of research for decades, both in the context of local and wide area networks (computer networks), and as an interconnect for parallel machines. Both are very much related to on-chip networks, and many of the results in those fields are also applicable on chip.”); Ex. 1009 (Jantsch) at 48 (native page 86) (“[M]any research groups have concurrently proposed the idea of using a packet switched communication network, similar to one used in computer networks, for on chip communication.”), 48-52 (native pages 86-95); Ex. 1034 (Duato) at 6 (native page vii) (“Originally developed for the demanding communication requirements of

multicomputers, interconnection networks are beginning to replace buses as the standard system-level interconnection.”).

37. Because NoCs can have more diverse types of communicating components beyond traditional parallel machines (e.g., DMA engines transferring data between I/O and on-chip memory, video streaming to frame buffers, CPU cache misses to memory, data transfers between CPUs and GPUS, etc.), they are often extended to also support quality-of-service (QoS) across different communication streams.

38. By mid-2004, the computer architecture research community was well-aware of NoCs. In addition to the Dally et al paper, there were a number of other research papers and books published on NoCs in 2001, 2002, and 2003. Ex. 1013 (Benini I); Ex. 1026 (Benini II); Ex. 1025 (Kumar); Ex. 1035 (Hemani); Ex. 1036 (Wiklund); Ex. 1009 (Jantsch).

C. Network Optimization

39. By mid-2004, it was well known that transferring messages of a larger size over a network, rather than transferring a higher volume of smaller-sized messages over the network, improves the network’s performance. Ex. 1018 (Culler) at 101 (native page 838) (“Making messages larger has several advantages.”) For example, it was well-known that there is overhead associated with each transmission, and using larger messages “amortizes the per-message endpoint

overhead over the large amount of data being sent.” Ex. 1018 (Culler) at 104 (native page 848) (stating “[m]aking messages large is important for amortizing overhead and for ensuring that the rate of the communication pipeline is not limited by the endpoint overhead of message processing”); Ex. 1030 (Heinlein 1994) at 43, 46, fig. 7 (stating that “chunking” data for sending over a network “is effective because it amortizes the overhead of starting the transfer handler and saving and restoring the transfer state over multiple lines” and correlating “performance” with “chunking” in Figure 7); Ex. 1019 (Heinlein 1997) at 20 (attributing a “25–40%” improvement in “transfer performance” to “chunking” due to “amortiz[ation of] the overhead of starting the transfer handler and saving and restoring the transfer state”). As another example, using larger messages can reduce the number of acknowledgements that need to be sent through the network, potentially providing large reductions in latency, traffic, overhead, and contention. Ex. 1018 (Culler) at 101 (native page 838) (“[A] large message may require only a single acknowledgment rather than one per word, which can reduce latency as well as traffic and contention”). As a further example, using larger messages could reduce start-up costs associated with the transfer time for moving data over a network. “[T]he start up cost” for transferring data—i.e., “the time for the first bit get to the destination”—was a known “constant term” added into the time to transfer data, which a POSITA would have understood is reduced when the amount of data associated with a single data transfer is increased

(thereby reducing the overall number of messages, each invoking its own start-up cost). Ex. 1018 (Culler) at 6 (native page 60).

VI. THE '800 PATENT

40. The '800 Patent issued on December 27, 2011, from U.S. Application No. 11/569,083 (Ex. 1004, "the '083 Application"), which was filed on Nov. 14, 2006. The '083 Application is a national stage application (under 35 U.S.C. § 371) of PCT/IB2005/051580, which was filed on May 13, 2005. The '800 Patent claims priority to EP Application No. 04102189, which was filed on May 18, 2004.

A. Claims

41. The '800 Patent has 21 claims, including four independent claims numbered 1, 10, 11, and 12. Claims 1, 2, 4, 6, 8–12, 13, and 18–21 are the Challenged Claims.

42. Exemplary Claim 12 is reproduced below:

12. An integrated circuit comprising:

- a plurality of processing modules including a first processing module having a first processing memory and a second processing module having a second processing memory;

- a plurality of interconnect modules including a first interconnect module and a second interconnect module, wherein the first processing module is connected to the first interconnect module and the second processing module is connected to the second interconnect module; and

- a network for connecting the first interconnect module to the second interconnect module in order to provide a communication connection between the first processing module and the second

processing module, wherein the first interconnect module is connected between the first processing module and the network, and wherein the second interconnect module is connected between the second processing module and the network;

wherein the first interconnect module includes a first memory for buffering first data from the first processing module, the first interconnect module further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first memory, and the second interconnect module includes a second memory for buffering second data from the second processing module, the second interconnect module further including a second determination unit, wherein the second determination unit is configured to determine a second optimal amount of data to be buffered by the second memory,

the first Interconnect module transferring the first data to the second processing module when the first data buffered in the first memory reaches the first optimal amount, and the second interconnect module transferring the second data to the first processing module when the second data buffered in the second memory reaches the second optimal amount

wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.

Ex. 1001, 11:44-12:31.

B. Summary of the Specification

43. The '800 Patent is titled "Integrated circuit and method for buffering to optimize burst length in networks on chips," and relates to "relates to an integrated circuit having a plurality of processing modules and an interconnect means for coupling said plurality of processing, a method for buffering and a data processing system." Ex. 1001 at Title, 1:5-10, 3:35-37, 4:29-36. The patent refers to its processing modules as "intellectual property blocks" (or "IPs"), which are further described as "master module M" and "slave module S." Ex. 1001 at 3:38-57. These processing modules communicate with one another via "a network on chip NOC." Ex. 1001 at 3:38-41, 5:10-11, 5:51-55. More specifically, the master and slave modules interact with each other over a network, and each connect to the network via "network interface[s] NI." Ex. 1001 at 3:38-57, 5:39-47, 5:55-57.

44. The '800 Patent further explains that the IP blocks send data over the network, and such data is buffered in memory of the network interfaces "until the conditions for an optimal transfer are satisfied." Ex. 1001 at 5:58-64, 6:15-17; *see also* Ex. 1001 at 7:23-25. When data is transmitted over the network, it is transmitted "as a packet within a certain burst length" that is "adopted to suit the network N in order to improve the network utilization." Ex. 1001 at 5:64-66, 6:4-6. The '800 Patent's network also uses "determination units" in the network interfaces that

determine when to send date (“the transmission moment”). Ex. 1001 at 6:47-53, 7:4-6; *see also* Ex. 1001, 7:7-14.

C. Summary of Prosecution History

45. I have reviewed the file history of the '800 Patent.

46. The '083 application was originally filed with fourteen claims directed to an integrated circuit, method of buffering data, and data processing system. Ex. 1004 at 8-13, 66. Throughout prosecution, the claims were repeatedly rejected as obvious over the prior art, Ex. 1004, 73-82, 119-34, 204-18, 258-74, 330-49, 410-29. The rejections led to some claims being cancelled, some claims being repeatedly narrowed through amendment, and other claims being added. Ex. 1004, 94-105, 147-57, 240-50, 308-19, 391-402, 475-87.

47. The application was allowed on August 26, 2011, with the examiner noting as the reason for allowance that “the prior art found alone or in combination does not teach or suggest that the communication properties of the communication between the master and slave used to determine the optimal moment for sending data, incorporate the claimed list of communication properties to make that determination.” Ex. 1004 at 653.

VII. LEVEL OF ORDINARY SKILL IN THE ART

48. I understand that there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including the educational level of active

workers in the field at the time of the alleged invention, the sophistication of the technology, the type of problems encountered in the art, and the prior art solutions to those problems.

49. In determining the characteristics of a hypothetical person of ordinary skill in the art of the '800 Patent at the time of the alleged invention, I considered several things, including the type of problems encountered in this field, and the rapidity with which innovations were made. I also considered the sophistication of the technology involved, and the educational background and experience of those actively working in the field, and the level of education that would be necessary to understand the '800 Patent. Finally, based on my own experience, I placed myself back in the relevant periods of time, and considered the state of the art and the level of skill of the engineers working in this field in those times.

50. It is my opinion that a POSITA at the time of the '800 Patent would have had at least a Bachelor's degree in computer science, electrical engineering, computer engineering, or an equivalent technical degree, and two or more years of experience with system-on-chip, interconnect, or networking technologies. Additional education might compensate for less experience, and vice-versa.

51. I understand that Patent Owner has proposed a somewhat different level of skill in the art in parallel district court proceedings:

NST's expert, Dr. Chmelar, opines that a person of ordinary skill in the art ("POSITA") at the time of the invention of the Asserted Patents is

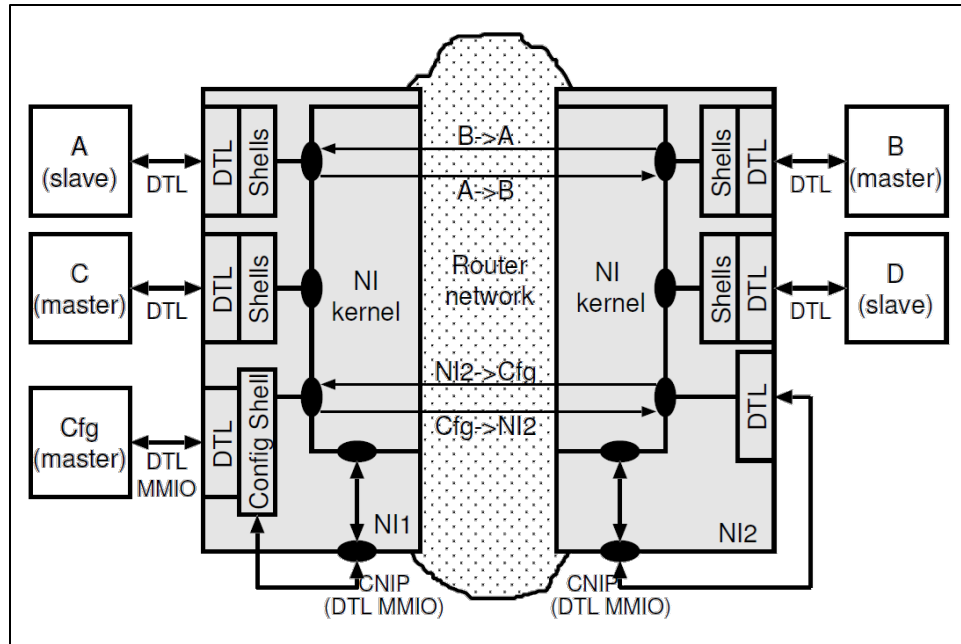
an individual having at least: a Bachelor of Science (or equivalent) degree in electrical engineering, computer engineering, or a related field, and 2-3 years of work experience in very large-scale integrated (“VLSI”) systems, such as application-specific integrated circuits (“ASICs”), application-specific standard parts (“ASSPs”), system-on-chip (“SoC”), or field-programmable gate arrays (“FPGAs”), and has an understanding of on-chip interconnection networks.

Ex. 1020 at 12 (native page 7). My opinions provided in this Declaration would not change under the level of ordinary skill in the art proposed by Patent Owner.

VIII. OVERVIEW OF THE ASSERTED PRIOR ART

A. Rădulescu

52. Rădulescu discloses “a network interface for an on-chip network,” referred to as an NoC, and in this particular case is called the *Æthereal* NoC. Ex. 1005 (Rădulescu) at 1. The NoC “is intended for systems on chip (SoC).” Ex. 1005 (Rădulescu) at 1. The structure of Rădulescu’s NoC is depicted in Figure 8 below. “IP modules,” such as master B and slave A, communicate with one another over a network (labelled “Router network” in Figure 8), via connections to network interfaces NI1 and NI2. Ex. 1005 at 3-4, FIG. 8.



Communication over the NoC “is performed using a transaction-based protocol, where master IP modules issue request messages ... that are executed by the addressed slave modules, which may respond with a response message” Ex. 1005 (Rădulescu) at 1, figs. 8–9. At each end of the communication, data messages are buffered in the corresponding network interface until “the amount of sendable data” satisfies “a configurable threshold,” at which point a “scheduler” determines the data can be transmitted over a router network. Ex. 1005 (Rădulescu) at 3–4, fig. 2. This threshold mechanism “optimize[s] the NoC utilization.” Ex. 1005 (Rădulescu) at 4.

B. Bahadiroglu

53. Bahadiroglu discloses “[a]n adaptive packet mechanism and method for optimizing data packet transmission through a network connection between a sending node and a receiving node.” Ex. 1006 (Bahadiroglu) at Abstract; [0047]–

[0048]. Bahadiroglu describes optimizing data transmission using “transmission characteristics or properties of a Connection 18.” Ex. 1006 (Bahadiroglu) at [0152], [0048]. Based on such conditions, Bahadiroglu’s adaptive packet mechanism operates to determine an “optimum packet size and an optimum inter-packet interval for transmission of packet data between the sending node and the receiving node” of a network. Ex. 1006 (Bahadiroglu) at [0048], [0077]. These optimum transmission characteristics are further adjusted “in real time” so that “operations of the neural network are continuously updated to reflect current conditions in the network.” Ex. 1006 (Bahadiroglu) at [0138], [0071].

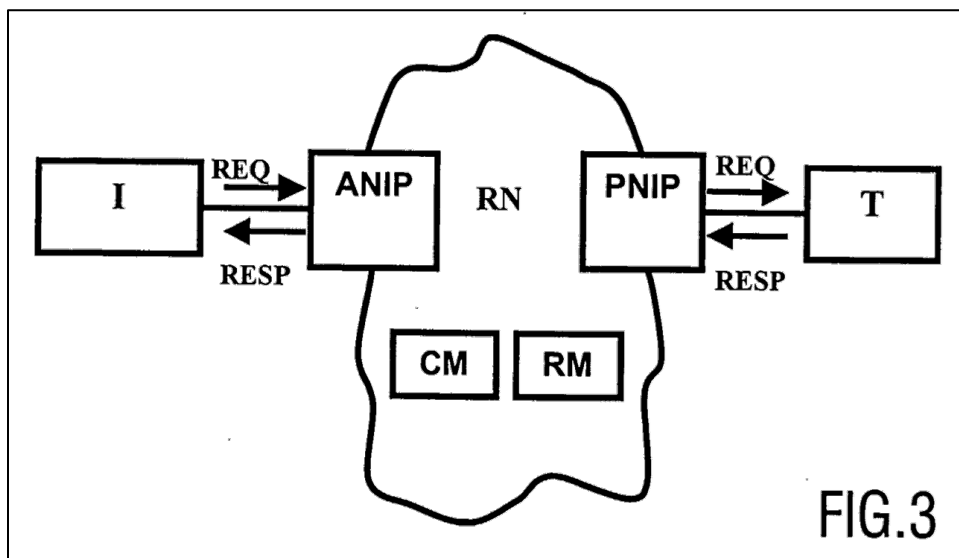
C. Petersen

54. Petersen discloses “an architecture for a network interface controller,” with a specific improvement to the “interfaces between communications networks and data processing systems.” Ex. 1007 (Petersen) at 1:16-21. Specifically, Petersen discloses using a “dedicated transmit buffer” to “automatically [initiate] transmission of a frame on the network when the number of bytes available in the transmit buffer memory exceeds a preprogrammed threshold.” Ex. 1007 (Petersen) at 3:61-64, 2:14–16, 2:31–34, 2:56–61. Petersen explains that the “amount of data of a frame which is downloaded by the host to the transmit buffer is monitored to make a threshold determination of an amount of data of the frame resident in the transmit buffer.” Ex. 1007 (Petersen) at 2:16–31, 2:49–53, 4:61–5:13; *see also* Ex.

1007 (Petersen) at Abstract. Further, Petersen’s “threshold value” is “dynamically programmable” based on “status information”—such as “indications of underrun conditions”—to optimize the threshold value and thus “optimize performance” of the host. Ex. 1007 (Petersen) at 2:31–34, 4:40–47, 4:56–60.

D. Rădulescu-WO

55. Rădulescu-WO is titled “Integrated Circuit And Method For Establishing Transactions” and discloses an “integrated circuit comprising a plurality of processing modules,” such as master and slave IP modules, and “a network arranged for providing connections between” those modules. Ex. 1008 (Rădulescu-WO) at Title, Abstract, 1:1–3, 5:7-8, 12:5–10. Those connections have a set of “properties” that may be used to efficiently adapt the connections to their requirements. Ex. 1008 (Rădulescu-WO) at 9:10–21, 11:5–22. An example of Rădulescu-WO’s integrated circuit is depicted in Figure 3.



IX. SPECIFIC GROUNDS FOR PETITION

A. Ground 1: Claims 1, 2, 4, 6, 8–12, 13, and 18–21 Are Rendered Obvious by Rădulescu in view of Bahadiroglu

56. In my opinion, claims 1, 2, 4, 6, 8–12, 13, and 18–21 are rendered obvious by the combination of Rădulescu and Bahadiroglu.

57. As a threshold matter, a POSITA would have been motivated to combine the teachings of Rădulescu with the teachings of Bahadiroglu in order to create a NOC with improved performance and efficiency, and would have had a reasonable expectation of success in doing so.

58. Rădulescu discloses “a network interface for an on-chip network” referred to as “Æthereal NOC.” Ex. 1005 (Rădulescu) at 2. Rădulescu’s NOC supports data communication between “IP modules,” such as communication between master and slave modules. Ex. 1005 (Rădulescu) at 1-2. Rădulescu recognizes that, “[t]o optimize the NoC utilization, it is preferable to send longer packets” of data. Ex. 1005 (Rădulescu) at 4. To achieve this optimization, Rădulescu further discloses using a “configurable threshold mechanism,” which determines when the “amount of sendable data” queued in the network interface is sufficiently large. Ex. 1005 (Rădulescu) at 4. By waiting to send data until the threshold is met, this mechanism produces longer packets that use the NoC more efficiently and consume less bandwidth to send a given amount of data. Ex. 1005 (Rădulescu) at 4. A POSITA would have understood that different, known ways of implementing

Rădulescu's configurable threshold existed at the time of the alleged invention. For example, the configurable threshold could be set with a fixed value (*e.g.*, a value programmed into a register), or it could be set with an adjustable value (*e.g.*, a value that changes at runtime according to conditions in the NoC). Ex. 1030 (Heinlein 1994) at 47 n.6 (teaching a mechanism for “dynamically adjust[ing]” an amount of stored data to send); Ex. 1018 (Culler) at 13 (native page 472) (describing use of registers as “output buffer[s],” which may optionally be configured for transactions “of fixed format” as opposed to “variable” format relating to buffering data “of variable size”). Rădulescu does not select a specific implementation for its configurable threshold mechanism.

59. Bahadiroglu is directed to the same goal of “optimizing data packet transmission through a network/connection,” Ex. 1006 (Bahadiroglu) at Abstract, [0047], and discloses a specific way to implement a threshold mechanism like the one disclosed by Rădulescu. Bahadiroglu discloses a mechanism for optimizing data packet transmission by using “adaptive control of data packet size.” Ex. 1006 (Bahadiroglu) at Title. Bahadiroglu describes that this adaptive mechanism “periodically determine[s]” the “current network conditions” and uses those conditions to “adjust[] the transmitted packet size ... in real time to that optimum for the current network and connection bandwidth” and/or other network properties. Ex. 1006 (Bahadiroglu), [0047]–[0048], [0071]. Bahadiroglu's mechanism also uses

the network conditions to “determine . . . an optimum inter-packet interval for transmission of data.” Ex. 1006 (Bahadiroglu), [0047]–[0048], [0071]. By determining the optimum interval between transmission of data, Bahadiroglu’s mechanism determines the optimum time to transmit each data packet. Further, the disclosed mechanism makes these determinations “in real time,” adjusting the “packet size and inter-packet interval [to] utilize[] the maximum available bandwidth of the connection most efficiently.” Ex. 1006 (Bahadiroglu), [0071]. The mechanism makes these real-time determinations based on various communication properties, including “the amount and type of data to be transmitted” and “network conditions” that include “current network latency, jitter and traffic conditions.” Ex. 1006 (Bahadiroglu), [0048], [0071], [0106]–[0107], [0111]–[0114].

60. A POSITA would have been motivated to combine Bahadiroglu’s adaptive control with Rădulescu’s network architecture to achieve the network-optimization benefits Bahadiroglu describes. More specifically, a POSITA would have been motivated to use Bahadiroglu’s “adaptive packet mechanism” to set the configurable threshold described by Rădulescu. A POSITA would have understood that using Bahadiroglu’s mechanism would improve the performance and efficiency of Rădulescu’s NoC because Bahadiroglu’s mechanism uses real-time network conditions to determine both packet sizing and transmission timing. A POSITA would have recognized that using this specific mechanism for Rădulescu’s

“configurable threshold mechanism” would increase the size of data packets being sent over the network to an optimum size based on network conditions, and as a result, reduce the number of transmissions required for a given amount of data, reduce the number of packet headers that must be sent, and ultimately allow more data to travel over the NoC in a more efficient manner. Bahadiroglu’s adaptive mechanism would also prevent the network from transmitting packets that are too large for current network conditions, which is a potential downside of implementing Rădulescu’s threshold mechanism using a fixed threshold value. Sending packets that are too large can result in buffers overflowing and packets being dropped. When packets are dropped they must be retransmitted, which is wasteful and inefficient. Similarly, optimizing the packet transmission timing based on current network conditions, as Bahadiroglu teaches, maximizes usage of the network bandwidth and manages traffic, increasing the likelihood of successful transmissions. As Bahadiroglu itself explains, “efficient bandwidth and traffic management is essential to obtain full advantage of the bandwidth and capacity of a network.” Ex. 1006 (Bahadiroglu) at [0011]; *see also* Ex. 1006 (Bahadiroglu) at [0011], [0013]–[0014]. A POSITA would have therefore understood that Bahadiroglu’s adaptive mechanism would help achieve Rădulescu’s stated goal of “optimiz[ing] the NoC utilization.” Ex. 1005 (Rădulescu) at 4. For those reasons, a skilled artisan would have been motivated to combine the teachings of Rădulescu and Bahadiroglu.

61. A POSITA would have further had a reasonable expectation of success in making such combination. A POSITA would have understood that combining Rădulescu and Bahadiroglu merely combines prior art elements (Rădulescu's configurable threshold mechanism and Bahadiroglu's adaptive packet mechanism) according to known methods (implementing Bahadiroglu's detailed disclosure of a specific type of configurable mechanism) to yield predictable results (network optimization). Moreover, both Rădulescu and Bahadiroglu are directed to the same field of optimizing data communication in a network. Ex. 1005 (Rădulescu) at 4; Ex. 1006 (Bahadiroglu) at [0046]–[0048], [0071]. A POSITA would have reasonably expected that using Bahadiroglu's known technique for optimizing data communication in a network through its adaptive packet mechanism would improve Rădulescu's NoC in the same way. It would require nothing more than routine implementation details to combine Bahadiroglu's mechanism with Rădulescu's NoC, and Bahadiroglu provides an extensive disclosure of such implementation details throughout its specification. As one example, Bahadiroglu discloses using either monitor packets or acknowledgements to determine and communicate the network conditions to a sending node so the sending node can use the network conditions as described above (to optimize packet size and timing). Ex. 1006 (Bahadiroglu) at [0049]. Bahadiroglu further describes the details of such methods using known network protocol stacks, including the Open Systems Interconnection

(OSI) Protocol Model, which Rădulescu likewise uses. Ex. 1005 (Rădulescu) at 1; Ex. 1006 (Bahadiroglu) at [0085].

62. A POSITA would also have understood that Bahadiroglu’s adaptive packet mechanism is directed to improving performance in a network (referred to as “network 10”), Ex. 1006 (Bahadiroglu) at [0073], which likewise applies to other networks that can benefit from such performance improvements, such as on-chip networks like that disclosed by Rădulescu. For example, Bahadiroglu’s adaptive packet mechanism is applicable to Rădulescu’s NoC because it would maximize bandwidth, capacity, and traffic management efficiencies in Rădulescu’s NoC, and optimize resources. *See* Ex. 1006 (Bahadiroglu) at [0011]; *see also* Ex. 1006 (Bahadiroglu) at [0011], [0013]–[0014]. As another example, Bahadiroglu describes an implementation that combines known analyzers to achieve a system that is relatively efficient, requires relatively little processing resources, and is relatively fast, which is likewise beneficial, and in fact, optimal, for on-chip services. Ex. 1006 (Bahadiroglu) at [0156]. A POSITA would have understood that combining Rădulescu and Bahadiroglu effectively balances NoC optimization with implementation cost and complexity.

63. Furthermore, as I explained in the technology background section above, NoCs have borrowed many design features from off-chip interconnection networks. For example, as described in one text devoted to networks-on-chip

“[d]esigning a network for on-chip communication is similar to the problem of designing a local area computer network with some computing and communication requirements.” Ex. 1009 (Jantsch) at 56 (native page 102); *see also* Ex. 1015 (Ye) at Abstract (“On-chip networks borrow features and design methods from those used in parallel computing clusters and computer system area networks”). Another chapter in the same text explained that addressing issues related to the optimization of network usage—as Rădulescu and Bahadiroglu do—is substantially similar in on-chip and off-chip networks: “The resource allocation, optimization of network usage and verification of performance and correctness are the main problems that are basically similar to what distributed and parallel system designers have to face.” Ex. 1009 (Jantsch) at 21 (native page 33). Indeed, the ’800 patent itself acknowledges that many of the issues faced by networks on chip had been studied in the context of off-chip networks and that results of those studies are applicable to on-chip networks as well. Ex. 1001 (’800 patent) at 1:48–64.

1. Claim 12

64. My analysis begins with claim 12 and then addresses the other challenged claims.

65. In my opinion, Rădulescu in view of Bahadiroglu teaches all the limitations of claim 12 and thus would have rendered it obvious.

a. Claim 12[preamble]: “An integrated circuit comprising”

66. I was not asked to analyze whether the preamble of Claim 12 is limiting. In any event, in my opinion, Rădulescu discloses an integrated circuit. Rădulescu “present[s] a network interface for an on-chip network” (*i.e.*, a NoC) and states that its network interface “is intended for systems on chip (SoC).” Ex. 1005 (Rădulescu) at Abstract, 1.

67. As I described above in the background of the technology (*see* Section V.A above) a POSITA would have understood that a SoC is implemented on a single integrated circuit chip. *See, e.g.*, Ex. 1009 (Jantsch) at 4 (native page vii) (“During the 1990s more and more processor cores and large reusable components have been integrated on a single silicon die, which has become known under the label System on Chip (SoC).”); Ex. 1023 (Wolf) at 5 (“An SoC is an integrated circuit that implements most or all of the functions of a complete electronic system.”); Ex. 1022 (Badawy) at 1 (“With the increase in the number of logic gates that can be implemented on a single chip, various functionalities ... can be integrated into a single silicon chip, realizing an entire system on chip (SoC)”); Ex. 1014 (Marinissen) at 1 (“Modern semiconductor process technologies enable the manufacturing of a complete system on one single die, the so-called system chip. Such system chips typically are very large ICs”); Ex. 1024 (Rincon) at 83-86 (describing the evolution of SoC components integrated onto a single chip).

68. A POSITA would have therefore understood that all of the components of Rădulescu's NoC, as well as the system-level components (e.g., IP modules) that interface with the NoC, would be implemented on the same integrated circuit (the SoC). For example, the entire system depicted in Rădulescu's Figure 8 is implemented on a single integrated circuit, or chip. Therefore, in my opinion, Rădulescu discloses an integrated circuit, including under the construction I understand that Patent Owner (PO) has proposed in parallel district court proceedings: "interconnected circuitry on a chip." *See* Ex. 2019 at 9.

69. Furthermore, the '800 Patent describes its alleged invention in the context of SoCs and similarly discloses "a network on chip" as an embodiment of the alleged invention. Ex. 1001 at 1:14–58, 4:66–5:11, 5:36–37, 6:7–8, figs. 1–2.

b. Claim 12[a]: "a plurality of processing modules including a first processing module having a first processing memory and a second processing module having a second processing memory"

70. In my opinion, Rădulescu discloses this limitation. Rădulescu discloses a plurality of processing modules called "IP modules," including "master IP modules" and "slave modules." Ex. 1005 (Rădulescu) at 1; *see also* Ex. 1005 (Rădulescu) at 2, 5, fig. 9. Rădulescu's Figure 8 depicts these IP modules, including "B (master)" (the claimed first processing module) and "A (slave)" (the claimed second processing module), which I have annotated in red below.

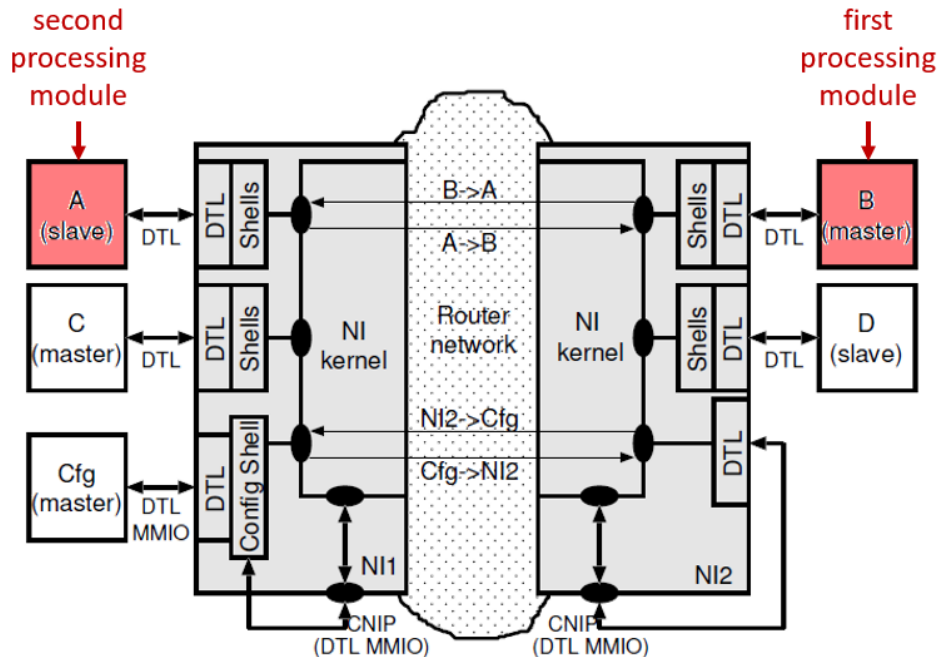


Figure 8. NI configuration

Ex. 1005 (Rădulescu) at fig. 8 (annotated).

71. A POSITA would have also understood that Rădulescu’s IP modules perform processing functions and include processing memory. Ex. 1005 (Rădulescu) at 1–4 (describing “IP modules”). Specifically, in the case of the SoC disclosed by Rădulescu, the IP modules perform “computation,” execute commands, and read and write data. Ex. 1005 (Rădulescu), 1–2. A POSITA would have understood that these functions involve processing and that Rădulescu’s IP modules therefore require local processing memory, such as registers and/or a cache, to perform these functions.

72. Using IP modules in SoC design was well-known at the time of the alleged invention. In particular, a POSITA would have understood that IP “modules” (also sometimes referred to as IP “blocks” or “cores”) are “pre-designed” such that

they can be used in a modular way to simplify and “enable a faster design cycle.” Ex. 1009 (Jantsch) at 101 (native page 193); Ex. 1012 (Goel) at 196 (“SOC design is typically a collection of Intellectual Property (IP) cores (possibly from independent vendors)”); Ex. 1022 (Badawy) at 2. Such IP modules that performed processing and included processing memory were well-known at the time of the alleged invention. For example, in Rădulescu-WO the author of Rădulescu explains that “so-called intellectual property blocks IPs” include “computation elements, memories or a subsystem.” Ex. 1008 (Rădulescu-WO) at 12:13–15; *see also* Ex. 1008 (Rădulescu-WO) at 6:7–9 (describing modules with “a memory, or a register”). As another example, Kumar describes that such IP blocks can comprise a “computation or storage unit or their combination,” such as “processor and DSP cores provided with caches as well as local memories.” Ex. 1025 (Kumar) at 3-4. Examples of such IP blocks are depicted in Kumar’s Figure 3 below.

synchronous. In Figure 3 RNI=resource network interface, P=processor core, D=DSP core, c=cache, M=memory and re=reconfigurable block.

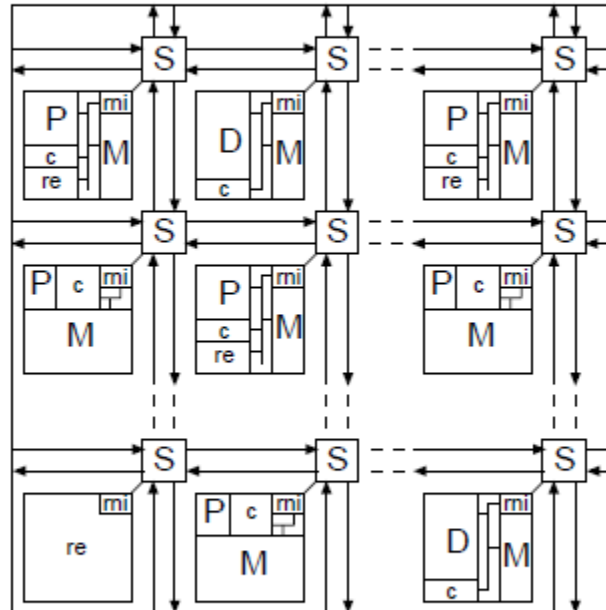


Figure 3. A typical NOC CLICHÉ featuring various types of resources.

Ex. 1025 (Kumar) at FIG. 3; *see also* Ex. 1016 (Lu) at 14 (“microprocessors usually have local caches,” “each node has its own local portion of the global main memory”), Fig. 2.5; Ex. 1026 (Benini II) at 5–6 (describing a CPU core with registers, local caches, and scratch-pad RAM), 8–9 (describing audio and video cores comprising processors, local caches, and local memory); Ex. 1015 (Ye) at 3 (native page 83) (“A typical shared-memory on-chip multiprocessor system ... consists of several node processors ... connected by an on-chip interconnect network. Each node processor has its own CPU/FPU and cache hierarchy (one or

two levels of cache).”), Fig. 2. Even basic IP modules typically include some type of memory, such as local caches, to store data and perform processing functions. *Id.*

73. Rădulescu’s disclosure of processing modules having processing memory is at least commensurate with the disclosure in the ’800 Patent, which describes that “the first processing module is a master” and “the second processing module is a slave,” and states “[t]he modules ... can be so-called intellectual property blocks IPs (computation elements, memories or a subsystem which may internally contain interconnect modules)” Ex. 1001 at 5:51–55, 8:48–51, 10:67–11:3; *see also* Ex. 1001 at 5:37–39 (disclosing “a master module M and a slave module S”), 7:35–40 (stating “the master M or the slave S ... may constitute intellectual property blocks”), 5:58–6:6, 6:37–38.

74. Therefore, in Rădulescu, each master IP module (including master B) is a first processing module having first processing memory and each slave IP module (including slave A) is a second processing module having second processing memory.

- c. **Claim 12[b]: “a plurality of interconnect modules including a first interconnect module and a second interconnect module, wherein the first processing module is connected to the first interconnect module and the second processing module is connected to the second interconnect module”**

75. In my opinion, Rădulescu discloses this limitation. Rădulescu explains that its NoC includes a plurality of “network interface[s] (NI)” that provide an

interface between the network and the IP modules, and enable the IP modules to communicate with each other through the network. Ex. 1005 (Rădulescu) at 1–4. For example, the “network interface (NI) is the component that provides the conversion of the packet-based communication of the NoC to the higher-level protocol that IP modules use.” Ex. 1005 (Rădulescu) at 3. This enables the disclosed “transaction-based” communication between IP modules, “where master IP modules issue request messages (e.g., read and write commands at an address, possibly carrying data) that are executed by the addressed slave modules, which may respond with a response message (i.e., status of the command execution, and possibly data).” Ex. 1005 (Rădulescu) at 1. Rădulescu’s Figure 8, annotated below, provides an example block diagram of the overall system, including a network interface NI2 (the claimed first interconnect module) connected to the “B (master)” IP module (the claimed first processing module) and a network interface NI1 (the claimed second interconnect module) connected to the “A (slave)” IP module (the claimed second processing module). Ex. 1005 (Rădulescu) at 5.

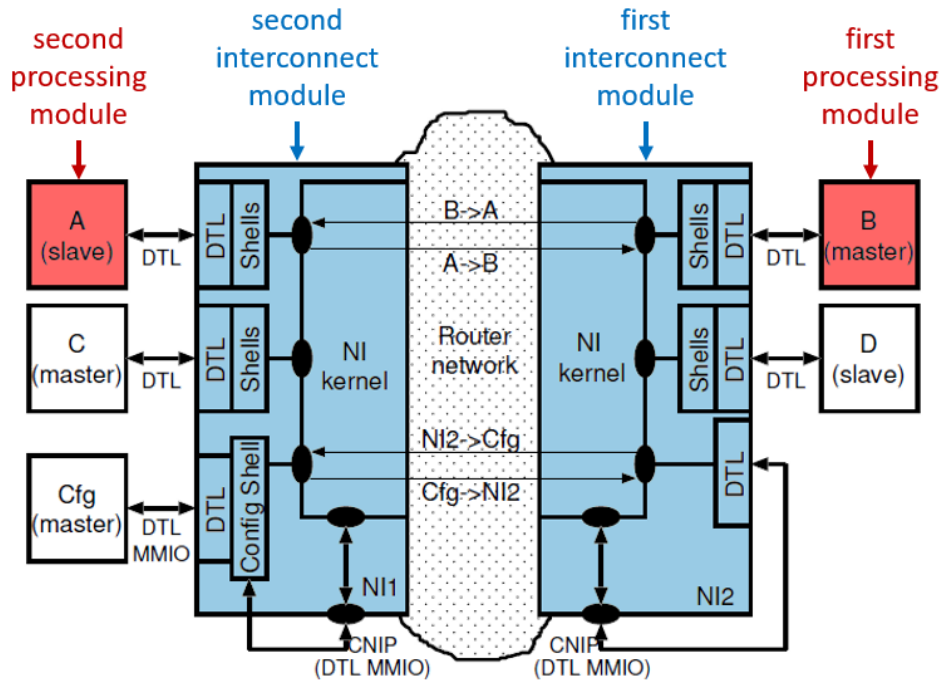


Figure 8. NI configuration

Ex. 1005 (Rădulescu) at fig. 8 (annotated); *see also* Ex. 1005 (Rădulescu) at figs. 1–6.

76. Rădulescu’s disclosure of network interfaces is identical (or nearly identical) to ’800 Patent’s disclosure of interconnect modules. For instance, the ’800 Patent discloses that each “master module M” and “slave module S” is “connected to a network” via “associated network interface[s] NI.” Ex. 1001 at 5:36–50; *see also* Ex. 1001 at 6:7–30 (similar), FIGS. 1–2.

- d. Claim 12[c]: “a network for connecting the first interconnect module to the second interconnect module in order to provide a communication connection between the first processing module and the second processing module, wherein the first interconnect module is connected between the first processing module and the network, and wherein the second interconnect module is connected between the second processing module and the network”**

77. In my opinion, Rădulescu discloses this limitation. For example, Rădulescu describes a network on chip (NOC). Ex. 1005 (Rădulescu) at 1, 3–5. Rădulescu further describes that this NoC comprises a network of routers for connecting network interfaces to provide a connection for transaction-based communications between master and slave IP modules. Ex. 1005 (Rădulescu) at 1, 3–5. For example, Rădulescu states “[n]etworks are composed of routers, which transport the data from one place to another, and network interfaces (NI), which implement the interface to the IP modules.” Ex. 1005 (Rădulescu) at 1. Rădulescu further explains that components in the network interfaces enable and facilitate “the IP modules” to communicate with one another via the router network. Ex. 1005 (Rădulescu) at 3, 6. Over the router network, Rădulescu’s IP modules (the claimed first and second processing modules) are permitted to communicate requests and responses to one another using channels connecting the NI interconnect modules, as Rădulescu describes and depicts. Ex. 1005 (Rădulescu) at 1, 5, FIG. 9.

78. Rădulescu's network, depicted in Figure 8 as the "router network" (annotated in green below), connects the first interconnect module NI2 to the second interconnect module NI1, which further provides a connection for the first processing module master B to communicate with the second processing module slave A (e.g., depicting "B -> A" and "A -> B" over the "router network," as shown in the red box). Ex. 1005 (Rădulescu) at 5. In other words, the network provides a communication network between the first and second processing modules, as shown below. Rădulescu also explains that these connections between processing modules "can be opened and closed at any time." Ex. 1005 (Rădulescu) at 2.

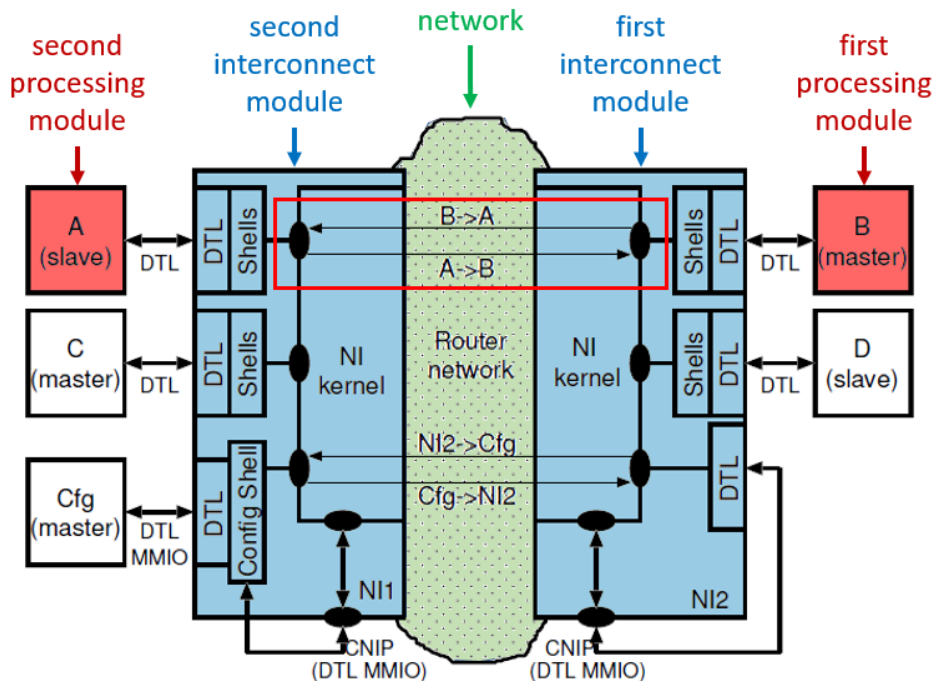


Figure 8. NI configuration

Ex. 1005 (Rădulescu) at FIG. 8 (annotated). In addition, Figure 8 also depicts that the network interface NI2 (the claimed first interconnect module) is connected

between master B (the claimed first processing module) and the network, and the network interface NI1 (the claimed second interconnect module) is connected between slave A (the claimed second processing module) and the network. Ex. 1005 (Rădulescu) at FIG. 8. Each network interface is therefore connected between an IP module and the network, and the network interfaces are both connected to the network. This configuration matches the configuration of the alleged invention depicted in the figures of the '800 Patent. *See* Ex. 1001 at FIGS. 1–2; *see also* Ex. 1001, 5:47–50 (disclosing a network of “routers” for routing data “from one network interface NI to another”).

79. I understand that Patent Owner has advocated in parallel district court proceedings for a construction of “network” as “a multi-hop interconnection between modules separated by one or more network nodes.” Ex. 1020 at 13 (native page 8). To the extent Patent Owner urges such a construction here, in my opinion, a POSITA would have understood that Rădulescu discloses such a network. Rădulescu’s interconnect modules are not directly connected, but rather are separated by at least one, and in some cases a “number of routers,” which are network nodes. Rădulescu further discloses that data passes through such “routers” to “reach its destination,” meaning that a data packet must traverse (or hop) through one or more routers as it travels from its source to its destination. Rădulescu therefore

discloses a “multi-hop interconnection between modules separated by one or more network nodes.” Ex. 1005 (Rădulescu) at 3-4; Ex. 1020 at 13 (native page 8).

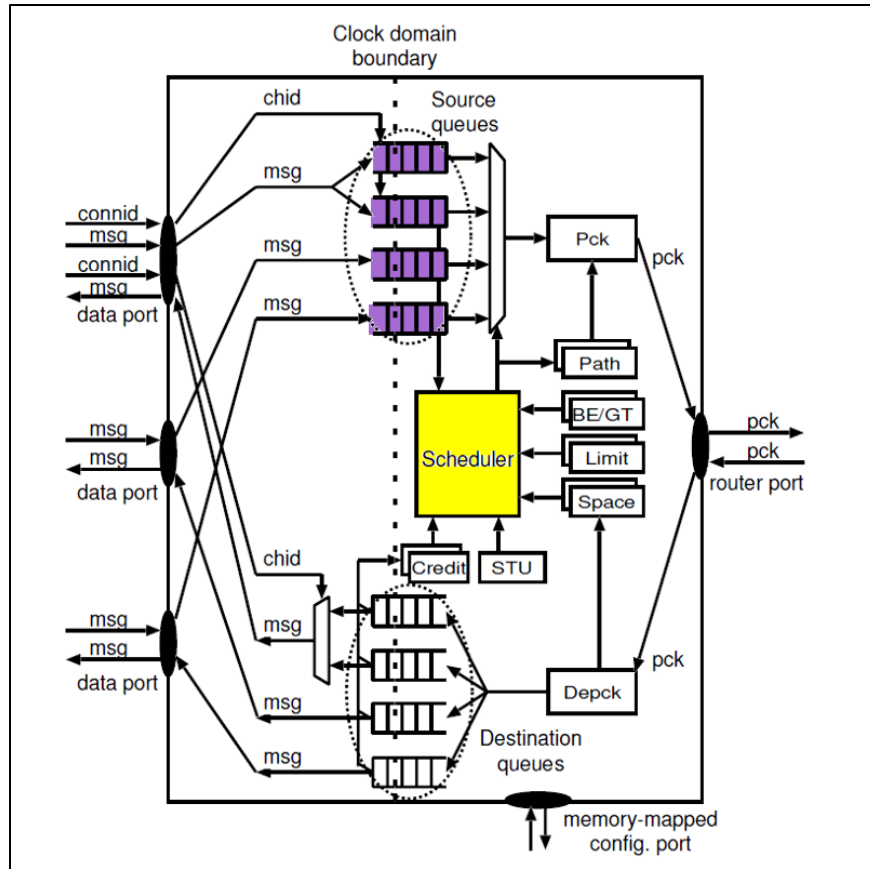
- e. **Claim 12[d]: “wherein the first interconnect module includes a first memory for buffering first data from the first processing module, the first interconnect module further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first memory, and the second interconnect module includes a second memory for buffering second data from the second processing module, the second interconnect module further including a second determination unit, wherein the second determination unit is configured to determine a second optimal amount of data to be buffered by the second memory”**

80. In my opinion, Rădulescu in view of Bahadiroglu renders this limitation obvious. *First*, Rădulescu discloses that each of its network interfaces NI2 and NI1 (the claimed first and second interconnect modules, respectively)¹ include memory.

¹ As explained in this ground, Rădulescu’s network interface NI2 is mapped to the claimed first interconnect module, and Rădulescu’s network interface NI1 is mapped to the claimed second interconnect module. However, these interfaces are symmetric and a POSITA would have understood that communication occurs in both directions such that mapping each interface (and the accompanying components and functionality that follows) to the opposite claim elements likewise renders this claim obvious. For the same reason, a POSITA would have

As Rădulescu explains, the NI kernels in each of the network interfaces NI2 and NI1 have “two message queues for each point-to-point connection (one source queue, for messages going to the NoC, and one destination queue, for messages coming from the NoC).” Ex. 1005 (Rădulescu) at 3. Rădulescu further explains that “memory [is] allocated for the queues.” Ex. 1005 (Rădulescu) at 1. This memory is annotated purple in Figure 2 of Rădulescu below.

found it obvious to apply Bahadiroglu’s disclosures (as described further below) to both interfaces, and both would benefit from Bahadiroglu’s teachings in the same way.



Ex. 1005 (Rădulescu) at FIG. 2 (annotated).

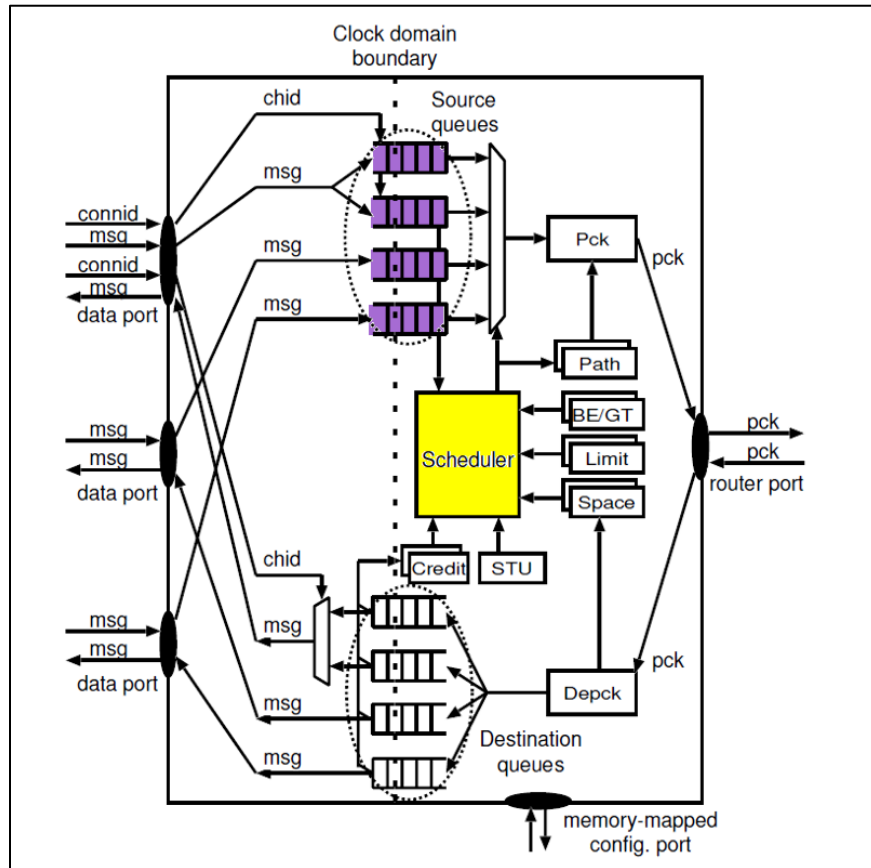
81. Rădulescu further explains that these “queues are area-efficient custom-made *hardware fifos*.” Ex. 1005 (Rădulescu) at 3, 5. A POSITA would have understood that hardware FIFOs were a well-known form of memory. *See, e.g.*, Ex. 1027 (Modern Dict. Of Elecs.) at 3 (“first-in, first-out (FIFO) memory — 1. type of memory with separate input and output ports”); Ex. 1028 (Penguin Dict. Of Elecs.) at 3 (“FIFO *Abrev. for* first in, first out, a type of memory system”); Ex. 1029 (Dict. Of Computer and Internet Terms) at 3 (“FIFO (first-in-first-out) ... a data structure or memory device from which items are retrieved in the order in which they were

stored.”). Rădulescu itself describes hardware FIFOs as an alternative to “RAMs,” or “random access memories.” Ex. 1005 (Rădulescu) at 5. Such FIFO memory is often used as a buffer. Ex. 1027 (Modern Dict. Of Elecs.) at 3. Buffers are used to temporarily store or stage data until it is ready to be processed or transmitted. Ex. 1030 (Heinlein 1994) at 39; Ex. 1009 (Jantsch) at 39 (native page 69). Buffers are often used to store smaller segments of data until enough segments exist in the buffer to be sent as a larger packet of data. *See* V.C.

82. ***Second***, Rădulescu discloses that the source queues in the NI kernels of the interconnect modules (the claimed first and second memory) buffer data from their associated IP blocks (the claimed first and second processing modules). As Rădulescu explains, “[t]he NI kernel ... receives and provides messages, which contain the data provided by the IP modules.” Ex. 1005 (Rădulescu) at 3. Once the data is received, the source queues “buffer” the data before it is transmitted. *See* Ex. 1005 (Rădulescu) at 4 (describing a flush mechanism to prevent “data being buffered indefinitely” in the source queues). A POSITA would have understood that Rădulescu’s disclosure is consistent with known techniques for buffering data in a source memory to facilitate network communications. Ex. 1018 (Culler) at 13 (native page 472).

83. ***Third***, Rădulescu discloses that each of its network interfaces NI2 and NI1 (the claimed first and second interconnect modules) include a scheduler. An

example of this scheduler is depicted in Figure 2 (annotated in yellow below), which is connected to the queues that buffer data (annotated in purple below).



Ex. 1005 (Rădulescu) at Fig. 2.

84. As Rădulescu explains, the scheduler determines whether there is “data which can be transmitted,” and if so, grants a certain channel permission to transmit the data. Ex. 1005 (Rădulescu) at 3. To make that determination, Rădulescu’s schedulers use “a configurable threshold mechanism” that defines a particular threshold amount of sendable data that must exist in a source queue before the data “can be transmitted.” Ex. 1005 (Rădulescu) at 3-4. The “scheduler[s] . . . select

connections from which messages are consumed, based e.g., on their filling.” Ex. 1005 (Rădulescu) at 4. That is, a filled “queue becomes eligible for scheduling when ... the amount of sendable data are above a first threshold,” and the scheduler will “skip[] a channel as long as the sendable data is below the threshold.” Ex. 1005 (Rădulescu) at 4. This threshold mechanism is therefore used to set the amount of data that will be buffered in the source queue before such data is transmitted.

85. I understand that Patent Owner has advocated in parallel district court proceedings for a construction of “determination unit” as “circuit that makes determinations based on stored or received information.” Ex. 1021 at 44 (native page 38).² A POSITA would have understood that, under that construction, the scheduler included in the NI2 (the claimed first interconnect module) and the scheduler included in the NI1 (the claimed second interconnect module) are each a “determination unit” included in each respective interconnect module at least because each scheduler makes a determination about whether to transmit buffered data based on a configurable threshold that is stored or received by the scheduler.

² I understand that other parties have taken the position that “determination unit” is a means-plus-function term lacking corresponding structure in the specification, but that patent owner disagrees and contends that this term is not a means-plus-function term. Ex. 1021 at 44 (native page 38).

Ex. 1005 (Rădulescu) at 4. The schedulers are also each a circuit. As Rădulescu explains, its network interface “is intended for systems on chip (SoC)” and is operationalized “by implementing the protocol stack in hardware.” Ex. 1005 (Rădulescu) at 1; *see also* Ex. 1005 (Rădulescu) at 2 (“In the Æthereal NoC, we optimize the performance and minimize the cost of the protocol stack by implementing it in hardware, rather than in software.”). A POSITA would have understood that such hardware is a circuit.

86. Rădulescu further describes that the threshold mechanism in its schedulers “optimize[s] the NoC utilization” because “it is preferable to send longer packets.” Ex. 1005 (Rădulescu) at 4. In this way, Rădulescu’s schedulers optimize the packet size, and thus determine an optimal amount of data³ to be buffered by

³ I understand Petitioner believes the term “optimal amount of data” is indefinite because the scope of what is “optimal” is subjective, while Patent Owner disagrees and has argued that this term would have been understood to “equate[] to the amount of data that is optimal under certain conditions.” Ex. 1020 at 34 (native page 29). I have not analyzed whether this term is definite. Regardless of the precise scope of “optimal amount of data,” in my opinion, it is clearly disclosed by Bahadiroglu, which expressly describes determining the “*optimum* packet size,” as discussed further below.

each respective memory, as I described above in Section IX.A, and as was widely known in the art. *See* V.C. Rădulescu’s optimization of the amount of data to be buffered is consistent with the ’800 patent, which states that “packets ... should be sufficiently large to be transferred in an optimal way with regard to the network N.” Ex. 1001 at 7:15–28.

87. A POSITA also would have found this element obvious in view of Bahadiroglu, which describes a determination unit configured to determine an optimal amount of data to be buffered in a network interface memory. Specifically, Bahadiroglu describes an adaptive packet mechanism that determines an optimum packet size in order to optimize the transmission of data through a network. Ex. 1006 (Bahadiroglu) at Abstract, [0047]–[0048]; *see also, e.g.*, Ex. 1006 (Bahadiroglu) at [0051]–[0052], [0054], [0077]–[0078], [0107], [0109], [0113], [0120]–[0121], [0127]–[0128], [0133], [0152]–[0153], [0163], [0164]. Bahadiroglu explains this as follows:

As will be described in the following, the method and mechanism of the present invention determines, in real time, the packet size and inter-packet interval that utilizes the maximum available bandwidth of the connection most efficiently under current network latency, jitter and traffic conditions by comparison of the relative latency and jitter of different sized packets. The present invention then adjusts the transmitted packet size and inter-packet interval in real time to that optimum for the current network and connection bandwidth.

Ex. 1006 (Bahadiroglu) at [0071].

88. A POSITA would have understood that, because data must be buffered before it can be assembled into packets, Bahadiroglu's determination of the "optimum packet size" is also a determination of the optimal amount of data that should be buffered before it is transmitted. As Bahadiroglu explains, data that is passed to network interfaces for transmission is "packetize[d]" into packets "of the size designated by the Optimum Packet Size." Ex. 1006 (Bahadiroglu) at [0119]–[0121] ("The Packet Transfer Engine (PTE) 20TS will form the Data 22D into Data Packets 16C of the size designated by Optimum Packet Size 36S...."), [0126], fig. 6A. A POSITA reading this disclosure in Bahadiroglu would have understood that buffering of data is necessary in order to packetize the data for transmission over the network. The connection between buffering and packetizing is disclosed in the '800 Patent, which treats the amount of data for buffering as equivalent the "optimal packet size." *See* Ex. 1001 at 6:49–65, 7:14–34 ("The data is buffered until the buffered data reach an optimal packet size such that the buffered data can be transferred over the network.").

89. Bahadiroglu also describes "updating the optimum packet size" in its nodes based on current network conditions. Ex. 1006 (Bahadiroglu) at [0047]–[0048], [0052]. Bahadiroglu's disclosure of determining an optimal amount of data in a sending node is a determination of a first optimal amount of data. Similarly, Bahadiroglu's disclosure of determining an optimal amount of data in a receiving

node is a determination of a second optimal amount of data. This is consistent with Rădulescu's disclosure that there are schedulers in both NI2 and NI1 that utilize a configurable threshold mechanism. Additionally, a POSITA would have understood that the act of updating these optimal packet sizes over time based on network conditions, as Bahadiroglu discloses, would further demonstrate (and thus teach) first and second optimal amounts of data.

90. As I described in Section IX.A and immediately above, a POSITA would have looked to Bahadiroglu's detailed disclosures of an adaptive packet mechanism as a way to configure Rădulescu's threshold mechanism, and would have had a reasonable expectation of success in configuring Rădulescu's mechanism as an adaptive packet mechanism. When combined in that way, Rădulescu's network interface NI2 (the claimed first interconnect module) includes a scheduler (the claimed first determination unit) that is configured to determine a first optimal amount of data to be buffered by the source queue in the network interface NI2 (the claimed first memory). Similarly, when combined, Rădulescu's network interface NI1 (the claimed second interconnect module) includes a scheduler (the claimed second determination unit) that is configured to determine a second optimal amount of data to be buffered by the source queue in the network interface NI1 (the claimed second memory). Rădulescu in view of Bahadiroglu therefore renders this limitation obvious.

- f. Claim 12[e]: “the first interconnect module transferring the first data to the second processing module when the first data buffered in the first memory reaches the first optimal amount, and the second interconnect module transferring the second data to the first processing module when the second data buffered in the second memory reaches the second optimal amount”**

91. In my opinion, Rădulescu in view of Bahadiroglu renders this limitation obvious.

92. As I discussed above in relation to claim element 12[d], Rădulescu describes “optimiz[ing] the NoC utilization” by transmitting data buffered in the source queues (the claimed first and second memories) of each of its network interfaces NI2 and NI1 (the claimed the first and second interconnect modules) to the destination queues of the other network interface when the amount of buffered data is above “a configurable threshold” (an optimal amount). Ex. 1005 (Rădulescu) at 4; *see also* Ex. 1005 (Rădulescu) at fig. 8 (showing data flowing through Rădulescu’s NoC from slave A to master B, and vice versa). The data from network interface NI2 (the claimed first interconnect module) is therefore transferred to slave A (the claimed second processing module) when the amount of buffered data in the memory of NI2 (the claimed first data buffered in the first memory) reaches an optimal amount. Similarly, the data from network interface NI1 (the claimed second interconnect module) is therefore transferred to master B (the claimed first

processing module) when the amount of buffered data (the claimed second data buffered in the second memory) reaches an optimal amount.

93. Likewise, once Bahadiroglu's adaptive packet mechanism determines the optimum packet size, Bahadiroglu describes transmitting packets of that size over the network. *See* Ex. 1006 (Bahadiroglu) at Abstract, [0048], [0071] ("The present invention then adjusts the transmitted packet size and inter-packet interval in real time to that optimum for the current network and connection bandwidth."), [0119] ("Packet Transfer Engines ... [will] ... communicate 'payload' Data Packets 16C of the optimum packet size"), [0113] (describing that its system "use these optimum packet size and inter-packet interval values during the data transfer mode"), [0119]–[0121], [0078]. By necessity, an amount of data sufficient to fill Bahadiroglu's packets of the optimum size must be buffered before those packets can be sent. Thus, the combination of Rădulescu and Bahadiroglu renders this limitation obvious.

- g. Claim 12[f]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave”**

94. In my opinion, Rădulescu discloses this limitation and Rădulescu in view of Bahadiroglu renders this limitation obvious.⁴ Rădulescu discloses that its schedulers (the claimed first and second determination units), are configured to determine the moment at which data in the source queues is transmitted, as the name “scheduler” implies, based on communication properties of the communication between “B (master)” module and “A (slave)” module, which are a master and slave, respectively, as claimed. Ex. 1005 (Rădulescu) at 3–4. Rădulescu discloses that its schedulers determine the moment of transmission based on a number of “communication properties,” which is a broad term that includes many different properties that the claim goes on to recite in Element 12[g]. *See* Section IX.A.1.h

⁴ I understand Petitioner believes the claimed “optimal moment” is indefinite, and Patent Owner disagrees. I have not analyzed whether these terms are definite. Regardless of the precise scope of these terms, in my opinion, this limitation is clearly disclosed by Rădulescu, and rendered obvious by Rădulescu in view of Bahadiroglu.

below. For example, Rădulescu describes that the schedulers determine the moment of transmission for data with “time guarantees (GT)” based on whether “the current slot is reserved for a GT channel,” which a POSITA would understand corresponds to the bandwidth (throughput) communication property, and indirectly, latency, Ex. 1005 (Rădulescu) at 3; Ex. 1018 (Culler) at 5-7 (native pages 59-61); *see also* Ex. 1005 (Rădulescu) at 2 (discussing providing “throughput, latency and jitter guarantees.”). Rădulescu further describes that the schedulers determine the moment of transmission for data on the GT channel based on whether “there is space in the channel’s destination buffer,” which corresponds to the “flow control” communication property that Rădulescu discusses. Ex. 1005 (Rădulescu) at 3. As another example, if those prior conditions are not satisfied, the schedulers will transmit data on a best effort (“BE”) channel “with data and remote space using some arbitration scheme: e.g., round-robin, weighted round-robin, or based on the queue filling,” which corresponds to at least the “flow control” communication property. Ex. 1005 (Rădulescu) at 3–4. Applying Patent Owner’s interpretation of this claim limitation, these transmission moments are each an optimal moment, as Patent Owner has argued,⁵ for sending data according to communication properties

⁵ I understand that Petitioner believes the claimed “optimal moment” is indefinite because the scope of what is “optimal” is subjective, while Patent Owner

of the communication between the master B and the slave A (the claimed master and slave). Further, in my opinion, a POSITA would have understood these disclosures in Rădulescu are consistent with the meaning of the word “scheduler,” which a POSITA would have understood implies that the unit decides the moment for sending data.

95. In fact, Patent Owner has identified in the '800 Patent disclosure similar to that in Rădulescu to argue that each of these transmission moments constitute an optimal moment for sending data based on communication properties. Ex. 1020 at 33 (native page 28) (citing Ex. 1001 at 7:4–14 (“Besides the optimal burst length determination, also the transmission moment may be determined in a similar way by the first and second determination means BLDU1, BLDU2. For a guaranteed throughput GT connection, data should wait for one of the slots reserved for its

disagrees and has argued that this term would have been understood “to be the ‘transmission moment’ (*i.e.*, the moment in which data [is] transmitted), where the moment is based on the recited connection properties, including at least throughput.” Ex. 1020 at 33 (native page 28). I have not analyzed whether this term is definite. Regardless of the precise scope of “optimal moment,” in my opinion, it is clearly disclosed by Rădulescu and rendered obvious by Rădulescu in view of Bahadiroglu.

connection. Using a ‘flush’-like signal/flag may force the sending of the data in advance. Alternatively, for a best effort BE connection, a round robin arbitration across connections in the NI can be used. A ‘flush’-like signal may force a temporarily higher priority for a connection.”).

96. Rădulescu further discloses that the data the scheduler will determine and send at an optimal moment is sent from “the source queues” of network interfaces NI2 and NI1 (the claimed first memory and second memory, respectively). Ex. 1005 (Rădulescu) at 3; *see* Section IX.A.1.e above. Under Patent Owner’s interpretation of the terms “said first wrapper” and “said second wrapper,” “the source queue” of network interface NI2 is said first wrapper and the source queue of network interface NI1 is said second wrapper.⁶

97. In addition, Rădulescu in view of Bahadiroglu renders this limitation obvious. Bahadiroglu teaches using its adaptive packet mechanism to determine an optimal moment for sending the packetized data based on network conditions and

⁶ I understand that Patent Owner has argued that the terms “said first wrapper” and “said second wrapper” refer to the previously-recited “said first memory” and “said second memory,” respectively. Ex. 1021 at 50 (native page 44). I do not take a position on Patent Owner’s argument, but rather apply Patent Owner’s interpretation.

communication properties, as I also discuss above in Section IX.A. For example, Bahadiroglu describes that its adaptive packet mechanism “optimiz[es] data packet transmission through a connection” by using “current network conditions in the connection” to “determine ... an optimum inter-packet interval for transmission of packet data” between the sending and receiving nodes. Ex. 1006 (Bahadiroglu) at [0047]–[0048]; *see also* Ex. 1006 (Bahadiroglu) at Abstract, [0048], [0067], [0071], [0077], [0120], [0126], [0152], [0154], [0160], [0164]. As Bahadiroglu explains, the “inter-packet interval” is “the interval between packets that are transmitted consecutively through the connection,” or the amount of time between packet transmissions. Ex. 1006 (Bahadiroglu) at [0068], [0106]–[0107]; *see also* Ex. 1006 (Bahadiroglu) at [0069] (describing “inter-packet delay of 10 msec” and “inter-packet delay of 20 ms”). A POSITA would have understood that by determining the optimum interval between transmission of data, Bahadiroglu’s mechanism determines the optimal moment to transmit each data packet. Bahadiroglu further describes determining such “optimum inter-packet interval” based on “the amount and type of data to be transmitted” and “network conditions” that include “latency and jitter of packet transmission between the sending node and receiving node,” “available bandwidth,” and “traffic conditions.” *See* Ex. 1006 (Bahadiroglu), [0048], [0071], [0106]–[0107], [0111]–[0114].

98. For these reasons and those discussed in Section IX.A, a POSITA would have found it obvious and would have been motivated to combine Rădulescu's disclosure of schedulers that determine when to transmit data buffered in network interface source queues based on communication properties, with Bahadiroglu's disclosure of an adaptive packet mechanism that determines the optimum inter-packet interval based on the same or similar network conditions, and would have had a reasonable expectation of success in doing so, as I explained above in Section IX.A. Thus, a POSITA would have understood that Rădulescu in combination with Bahadiroglu teaches that the determination units in either or both of Rădulescu's network interfaces NI1 and NI2 (the claimed interconnect modules) are configured to determine an "Optimum Inter-Packet Interval" as disclosed in Bahadiroglu (the claimed optimal moment) for sending the data according to communication properties of the communication between Rădulescu's master B and its slave A (the claimed master and slave).

- h. Claim 12[g]: “wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery”**

99. In my opinion, Rădulescu discloses this limitation and Rădulescu in view of Bahadiroglu renders this limitation obvious.⁷

100. Rădulescu discloses that many of the listed communication properties are used in the operation of its NoC. Ex. 1005 (Rădulescu) at 1, 6. Specifically, Rădulescu discloses using properties such as ordering of data transport, flow control, throughput, latency, data delivery, transmission termination, and transaction

⁷ I understand Patent Owner has argued that the determination unit need not use all of the listed properties in making its determination. Ex. 1021 at 41-44 (native pages 35–38). The claim appears to require the use of all listed properties. Applying Patent Owner’s interpretation, Rădulescu and Bahadiroglu alone and in combination disclose using one or more of the listed properties, as I discuss in this section. Further, it would have been obvious to use all listed properties, as I also discuss in this section.

completion in the operation of its NoC. For example, Rădulescu describes that “[s]upport for real-time communication is achieved by providing throughput, latency, and jitter *guarantees*.” Ex. 1005 (Rădulescu) at 2; *see also* Ex. 1005 (Rădulescu) at 1 (disclosing a “guarantee[]” on the “lower bounds on throughput” and “upper bounds on latency”), 6. Rădulescu also describes use of flow control: “End-to-end flow control ensures that no data is sent unless there is enough space in the destination buffer to accommodate it.” Ex. 1005 (Rădulescu) at 3; *see also* Ex. 1005 (Rădulescu) at 6. Rădulescu’s disclosure of flow control matches the claim language, as well as the construction proposed in district court litigation. Ex. 1011 at 13 (native page 6); *see also* Ex. 1033 at 279-280. As another example, Rădulescu describes ordering of data transport. *See* Ex. 1005 (Rădulescu) at 2 (describing “order or un-ordered message delivery”), 4 (describing “[i]n-order delivery” of data), 6 (stating the NoC offers “transaction ordering”). Rădulescu also describes “guaranteed message delivery or not” and data transmission “with or without timing guarantees,” which correspond to the properties of data delivery, transmission termination, and transaction completion. *See* Ex. 1005 (Rădulescu) at 2. For example, providing guaranteed delivery and timing is a form of data delivery and also provides information about transmission termination and transaction completion. Further, Rădulescu describes using a scheduler to “arbitrate” among transmission channels and using various “arbitration schemes,” such as “round-

robin, weighted round robin, or based on the queue filling,” which all correspond to the “priority” property. *See* Ex. 1005 (Rădulescu) at 3-4.

101. As discussed above in Section IX.A.1.g, a POSITA would have understood that Rădulescu’s schedulers consider these communication properties in determining when to transmit data buffered in the source queues of Rădulescu’s NI2 and NI1, meaning the schedulers determine the optimal moment to send that data according to these communication properties.

102. In addition, Bahadiroglu describes determining the “optimum inter-packet interval” (the claimed optimal moment) based on many of the same “communication properties,” which Bahadiroglu calls “Network Conditions.” Ex. 1006 (Bahadiroglu) at Abstract, [0001], [0047]–[0048], [0071]. For example, Bahadiroglu describes using information relating to bandwidth (also referred to as throughput, including in Bahadiroglu, *see* Ex. 1006 at [0024]), jitter, latency, “two way delay time,” “number of packets out of sequence,” and “number of packets lost” to determine the optimum inter-packet interval. *See* Ex. 1006 (Bahadiroglu), Abstract, [0048], [0049]–[0051], [0065], [0129]–[0132]; *see also* Ex. 1006 (Bahadiroglu) at FIG. 6A (listing network conditions). Bahadiroglu also explains that properties related to lossiness, data correctness, and data delivery, “such as packet error, packet loss, packet dropping, or packet retransmission [*sic*] rates[,] ... may be used to refine the results” and therefore factor into the determination of an

optimum inter-packet interval. Ex. 1006 (Bahadiroglu) at [0066]. As an example, Bahadiroglu describes considering information related to flow control. Ex. 1006 (Bahadiroglu) at [0013], [0093]–[0094], [0102]–[0103].

103. Rădulescu and Bahadiroglu—each independently and in combination—disclose determining the moment of transmission based on one or more of the claimed “communication properties.” In addition, considering all of the listed properties in scheduling data transmissions would have been obvious because the listed properties were well-known to POSITAs at the time of the alleged invention and it was similarly well-known to use them for this purpose. For example, Bahadiroglu expressly recognizes “that Network Conditions 42 directly or indirectly represent a significant number of network variables,” Ex. 1006 (Bahadiroglu) at [0153], and that Bahadiroglu’s adaptive packet mechanism may take into account “some or all of [the] Network Conditions 42.” Ex. 1006 (Bahadiroglu) at [0155].

104. Similarly, Rădulescu-WO, a patent publication by some of the authors of Rădulescu, that pertains to the very same network on chip technology discloses using all of the listed properties for the very purpose of scheduling data transmissions. For example, Rădulescu-WO teaches the same exact list of communication properties, referring to them as “connection properties”:

“[t]he connection properties may include ordering (data transport in order), flow control (a remote buffer is reserved for a connection, and a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data), throughput (a lower

bound on throughput is guaranteed), latency (upper bound for latency is guaranteed), the lossiness (dropping of data), transmission termination, transaction completion, data correctness, priority, or data delivery.”

Ex. 1008 (Rădulescu-WO) at 11:17–22, Abstract, 9:5–7, 9:18–10:12, 16:32–34.

Other papers by the same authors also describe these communication properties.

See, e.g., Ex. 1010 (Goossens) at 69 (describing “correctness and integrity of the result,” “promise of completion or delivery,” as well as “completion time (when is the result available), cumulative time to completion and its variations (e.g. bounded latency and jitter), and (peak and average) energy consumption”), 72–74 (describing data integrity, transaction ordering, transaction completion, flow control, throughput, latency, and jitter), Ex. 1033 (Rădulescu II) at 291–296 (describing data integrity, transaction completion, transaction ordering, latency, throughput, jitter, and flow control).

105. In my experience, all of these properties had long been known and used in data transmission scheduling before the time of the alleged invention. A POSITA would have been motivated to use all these network conditions in combination with Rădulescu and Bahadiroglu to further optimize data communication over Rădulescu’s NoC, and a POSITA would have had a reasonable expectation of success using such network conditions given that they were known and commonly used for that very purpose. For example, a POSITA would have understood that using such network conditions to optimize communication would maximize efforts

to control bandwidth, capacity, and traffic management over Rădulescu's NoC by accounting for the maximum number of possible factors that contribute to ensuring that communication between IP modules is effective and efficient. Using such network conditions to determine communication timing minimizes the potential for conflicts among traditional network efficiency indicators and traffic management levers that would otherwise degrade the transmission of data. In other words, by using a greater number of conditions (and the information that those conditions provide) than just one or a smaller number of factors, all relevant conditions are balanced and certain factors are not given outsized weight compared to others. Accounting for all factors is therefore the most effective way to accomplish Rădulescu's stated goal of "optimizing the NoC utilization." Ex. 1005 (Rădulescu) at 4.

2. Claim 13: "The integrated circuit of claim 12, wherein the plurality of interconnect modules manages the communication connection freeing the plurality of processing modules to perform data processing."

106. Claim 12 is obvious as I discussed above. *See* Section IX.A.1. In addition, in my opinion, Rădulescu discloses the limitation of claim 13 as it explains that its network interfaces—including NI2 and NI1 (the claimed interconnect modules)—manage communication over the NoC. *See, e.g.*, Ex. 1005 (Rădulescu) at 2 ("Packetization is performed by the NI, and is thus transparent to the IP modules."), 3 ("The network interface (NI) ... implement[s] the connections"), 3-4

(describing a “scheduler” that offloads a communication burden from the IP modules), 5 (“NI2 can be configured to set up a channel from B to A”). As a result, communication is “decoupled” from the computation performed by the processing modules. Ex. 1005 (Rădulescu) at Abstract (“Our network interface decouples computation from communication”), 1 (“Our NI ... is a key ingredient in achieving the decoupling between computation and communication”), 2. This is identical to the ’800 patent’s disclosure related to this claim limitation. Ex. 1001 at 1:42–47 (“NoCs also decouple computation from communication, which is essential in managing the design of billion-transistor chips. NoCs achieve this decoupling because they are traditionally designed using protocol stacks, which provide well-defined interfaces separating communication service usage from service implementation.”).

3. Claim 18: “The integrated circuit of claim 12, wherein the first processing module provides a first flushing command to flush the first memory and a second flushing command to flush the second memory so that the first data and the second data are sent without waiting to reach the first amount and the second amount.”

107. Claim 12 is obvious as I discussed above. *See* IX.A.1, above. In my opinion, Rădulescu also renders obvious the additional limitation of claim 18. Rădulescu discloses using “a flush signal for each channel (and a bit in the message header) to temporarily override the threshold” mechanism that buffers data, such that the flush signal “prevent[s] starvation at user/application level (e.g., due to write data

being buffered indefinitely ...).” Ex. 1005 at 4. A POSITA would have understood that this flush signal is a first flushing command that is provided by the master IP module (the claimed first processing module) to flush buffered data in the master source queue (the claimed first data in the first memory) so that the buffered data is sent before the threshold is reached (meaning, without waiting to reach the first amount, as claimed). *See also* Ex. 1005 at 2 (stating master IP modules “initiate transactions by issuing requests,” which includes “commands”). Rădulescu similarly explains that when the “flush signal is high for a cycle,” meaning the flushed command is enabled, “a snapshot of its source queue filling is taken, and as long as all the words in the queue at the time of flushing have not been sent, the threshold for that queue is bypassed,” such that the data in the queue is sent without waiting for the threshold to be reached. Ex. 1005 at 4. Rădulescu’s disclosure of a flush signal matches the ’800 patent’s disclosure, which explains that a “flush signal” can be used “to flush the buffered data” to prevent “deadlock caused by data being buffered” indefinitely. Ex. 1001 at 6:25–27, 7:9–10.

108. Rădulescu also discloses that a second “flush signal” may be used “to force credits to be sent even when they are below their threshold,” where the “credits are sent to the producer of data” to “indicate that more empty space is available” at the destination, such that this second flush signal allows more data to be sent from the source queue and therefore “prevent[s] possible starvation.” Ex. 1005 at 3–4.

This second flush signal is therefore a second flushing command that flushes credit data (such as “piggyback credits”) and the data such credits are attached to in the slave source queue (the claimed second data in the second memory) so that such credit data and the underlying data are sent before the threshold is reached (the claimed without waiting to reach the second amount). A POSITA would have also understood that this second flushing command is provided by the master IP module (the claimed first processing module) to force credits to be sent from the second interconnect module to the master interconnect module to indicate room is available at the second interconnect module for the data flushed as a result of the first flushing command. Ex. 1005 at 3–4.

4. Claim 19: “The integrated circuit of claim 18, wherein the first processing module provides the first flushing command and the second flushing command at least one of independently and simultaneously.”

109. Claim 18 is obvious as I discussed above. *See* IX.A.3, above. In my opinion, Rădulescu also renders obvious the additional limitation of claim 19. In terms of the relationship by which Rădulescu’s master IP module (the claimed first processing module) “provides” the first and second flushing commands, a POSITA would have understood that there are only a limited number of such relationships that can exist between two transmitted signals. Namely, such signals could be sent independently or dependently, and they could be sent simultaneously (at the same time) or sequentially (at different times). A POSITA would have therefore found it

obvious for Rădulescu's flush signals to be at least independent or simultaneous, particularly given the limited number of options for relating two signals. With respect to a NoC like Rădulescu's in particular, Rădulescu teaches and a POSITA would have understood that communication disruptions occur. Ex. 1005 at 4 (referring to "starvation" from "indefinite[]" buffering); Ex. 1033 at 278-79 (discussing reliability issues, deadlock and packet dropping), 292 (discussing transmission errors and failures). A POSITA would have understood a common remedial functionality for such event is to employ multiple commands simultaneously in order to restore communication as quickly as possible. In view of Rădulescu's disclosure of multiple flushing commands, employing a simultaneous approach to a data transmission disruption would include initiating both the first and second flushing commands at the same time, and a POSITA would have found such approach obvious. Ex. 1005 at 3-4.

5. Claim 20: "The integrated circuit of claim 18, wherein the second flushing command includes an indication to flush the second memory so that the first interconnect module forwards the second flushing command to the second interconnect module for flushing the second data from the second memory."

110. Claim 18 is obvious as I discussed above. *See* IX.A.3, above. In my opinion, Rădulescu also renders obvious the additional limitation of claim 20. As explained with respect to Claim 18, Rădulescu's master IP module (the claimed first processing module sends a second "flush signal ... [that] force[s] credits to be sent

even when they are below their threshold.” Ex. 1005 (Rădulescu) at 4. The credits are “piggybacked on packets” and indicate that space is available at a destination queue such that data may be transmitted to that queue. Ex. 1005 (Rădulescu) at 3-4; *see also* Ex. 1005 (Rădulescu) at 3 (“[W]e “piggyback credits in the header of the packets for the data in the other direction to improve NoC efficiency.”). A POSITA thus would have understood that the second flushing command is transmitted by the master IP module to the network interface NI1, such that the network interface NI2 (the first interconnect module) forwards the command to NI1 (the claimed second interconnect module) for flushing data from the source queue (the claimed second memory) so that “credits are sent to the producer of data” to indicate space is available at the NI1 interconnect module for the flushed first data from the first memory. A POSITA would have further understood that, because this flushing process stems from simultaneous commands issued at the master IP module, Rădulescu teaches the first interconnect module NI2 forwards the second flushing command to the second interconnect module NI1 for flushing the second data from the second memory (*i.e.*, from the slave NI1 source queue). Ex. 1005 (Rădulescu) at 3-4.

6. Claim 21: “The integrated circuit of claim 12, wherein the first processing module includes flush information in a header of a message to cause all messages from a same connection that have been sent earlier to be flushed from at least one of the first memory and the second memory.”

111. Claim 12 is obvious as I discussed above. *See* IX.A.1, above. In my opinion, Rădulescu also renders obvious the additional limitation of claim 21. As discussed above, Rădulescu discloses that its first processing module sends a flush signal to flush first data buffered in the first memory. *See* IX.A.3, above. Rădulescu also explains that this flush signal is included with “each channel” and includes “a bit in the message header” to flush “all the words [currently] in the queue.” Ex. 1005 at 4, FIG. 2. This data constitutes all messages from a same connection that have been sent earlier because this is the buffered data awaiting to be transferred. Ex. 1005 at 4. A POSITA would have therefore understood that the flush signal (including the flush information in the header of the message) causes all messages from a same connection that have been sent earlier to be flushed from the first memory.

7. Claim 10

112. In my opinion, Rădulescu in view of Bahadiroglu teaches all the limitations of claim 10 and thus would have rendered the claim obvious.

- a. **Claim 10[pre]: “A method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions”**

113. I was not asked to analyze whether the preamble of Claim 10 is limiting.

In any event, in my opinion, Rădulescu discloses a method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions. *See* Sections IX.A.1.a – IX.A.1.d.

- b. **Claim 10[a]: “configuring the first processing module having a first memory as a master the provides requests”**

114. *See* IX.A.1.b, above.

- c. **Claim 10[b]: “configuring the second processing module having a second memory as a slave the provides responses to the requests”**

115. *See* IX.A.1.b, above.

- d. **Claim 10[c]: “connecting the master to a master interface unit of the interface units”**

116. *See* IX.A.1.c, above.

- e. **Claim 10[d]: “connecting the master interface unit to the interconnect so that the master interface unit is between the master and the interconnect”**

117. *See* IX.A.1.d, above.

f. Claim 10[e]: “connecting the slave to a slave interface unit of the interface units”

118. *See* IX.A.1.c, above.

g. Claim 10[f]: “connecting the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect”

119. *See* IX.A.1.d, above.

h. Claim 10[g]: “determining by a master determination unit of the master interface unit a first optimal amount of data to be buffered by a master wrapper of the master interface unit”

120. *See* IX.A.1.e, above. Rădulescu discloses that each network interface has both source and destination queues, as shown for example in Figure 2, and data is buffered in each queue. *See* Section IX.A.1.e. In my opinion, a POSITA would have understood that source and destination queues in Rădulescu’s network interface NI2 (the claimed master interface unit) collectively constitute a master wrapper. Ex. 1005 (Rădulescu) at 3–4, FIGS. 2, 8; *see also* Ex. 1001 at 5:60–65 (stating “[t]he wrapper means” buffers data).

i. Claim 10[h]: “determining by a slave determination unit of the slave interface unit a second optimal amount of data to be buffered by a slave wrapper of the slave interface unit”

121. *See* IX.A.1.e, above. Rădulescu discloses that each network interface has both source and destination queues, as shown for example in Figure 2, and data is buffered in each queue. *See* Section IX.A.1.e. In my opinion, a POSITA would

have understood that source and destination queues in Rădulescu's network interface NI1 (the claimed slave interface unit) collectively constitute a slave wrapper. Ex. 1005 (Rădulescu) at 3–4, FIGS. 2, 8; *see also* Ex. 1001 at 5:60–65 (stating “[t]he wrapper means” buffers data).

- j. **Claim 10[i]: “buffering by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered”**

122. *See* IX.A.1.e, above.

- k. **Claim 10[j]: “transferring the buffered data from the slave wrapper to the master wrapper when said first optimal amount of data has been buffered by the slave wrapper”**

123. *See* IX.A.1.f, above.

- l. **Claim 10[k]: “buffering by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper”**

124. *See* IX.A.1.e, above.

- m. **Claim 10[l]: “transferring the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper,”**

125. *See* IX.A.1.f, above.

- n. **Claim 10[m]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave,”**

126. *See* IX.A.1.g, above.

- o. **Claim 10[n]: “wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

127. *See* IX.A.1.h, above.

8. Claim 1

128. In my opinion, Rădulescu in view of Bahadiroglu teaches all the limitations of claim 1 and thus would have rendered the claim obvious.

- a. **Claim 1[pre]: “An integrated circuit comprising:”**

129. *See* IX.A.1.a, above.

- b. **Claim 1[a]: “a plurality of processing modules being connected with an interconnect and each of the plurality of processing modules having a memory, wherein a first processing module communicates to a second processing module using transactions, wherein the first processing module is a master for providing requests and the second processing module is a slave for providing responses to the requests;”**

130. *See* IX.A.7.a–IX.A.7.c, above.

- c. **Claim 1[b]: “a first interface unit coupled to the slave and including a first wrapper for buffering data from the slave to be transferred over the interconnect to the master, the first interface unit further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first wrapper, and wherein the buffered data are transferred when said first optimal amount of data has been buffered by the first wrapper, and wherein the first interface unit including the first wrapper and the first determination unit is connected between the slave and the interconnect; and”**

131. *See* IX.A.7.f–IX.A.7.g, IX.A.7.i–IX.A.7.k, above.

- d. **Claim 1[c]: “a second interface unit coupled to the master and including a second wrapper for buffering data from the master to be transferred over the interconnect, the second interface unit further including a second determination unit, wherein second determination unit is configured to determine a second optimal amount of data to be buffered by the second wrapper, and wherein the buffered data are transferred when said second optimal amount of data has been buffered by the second wrapper, and wherein the second interface unit including the second wrapper and the second determination unit is connected between the master and the interconnect,”**

132. *See* IX.A.7.d–IX.A.7.e, IX.A.7.h, IX.A.7.l–IX.A.7.m, above.

- e. **Claim 1[d]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

133. *See* IX.A.7.n–IX.A.7.o, above.

9. **Claim 2: “The integrated circuit according to claim 1, wherein the first interface unit includes a first unbuffer-flag unit for providing a first unbuffer signal and the second interface unit includes a second unbuffer-flag unit for providing a second unbuffer signal, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to the first unbuffer signal and the second unbuffer signal, respectively, even if less than said first optimal amount and said second optimal amount of data is buffered in said first wrapper and said second wrapper.”**

134. Claim 1 is obvious as I discussed above. *See* IX.A.8, above. In my opinion, Rădulescu also renders obvious the additional limitation of claim 2. As discussed for claim 12, Rădulescu’s network interface units each include an NI kernel that “receives and provides messages ... contain[ing] the data provided by the IP modules.” Ex. 1005 at 3, FIGS. 2, 8. As also explained above with respect to claims 18 and 19, Rădulescu further explains that these messages include “a *flush* signal for each channel (and a bit in the message header) to temporarily override the threshold” for sending buffered data from the source queues before the threshold amount of sendable data is met, as well as another “flush signal to force credits to be sent [from the other side of the interconnect] even when they are below their threshold” to show space is available for data. Ex. 1005 at 4; *see also* Ex. 1004 (noting “credits are piggybacked on packets” of data). These flush signals are first and second unbuffer signals, respectively. That is, the NI kernels are first and second unbuffer-flag units for providing first and second unbuffer signals, and in response

to such signals, the source queue of each NI interface (said first and second wrappers) transfer buffered data in response to the unbuffer signals. By definition, when such buffered data is sent in response to an unbuffer signal, the buffered data in each queue is less than the first and second optimal amounts (otherwise it would have already been transmitted).

135. More specifically, a POSITA would have therefore understood that the slave network interface's NI kernel is a first unbuffer-flag unit for providing a first unbuffer signal (for example, either the flush signal to send data from the first wrapper in the slave NI1, or the flush signal to send data credits from the first wrapper in the slave NI1) and the master network interface's NI kernel is a second unbuffer-flag unit for providing a second unbuffer signal (for example, the flush signal to send data from the second wrapper in the master NI2). Ex. 1005 (Rădulescu) at 3-4; *see* IX.A.5, above.

10. Claim 4: “The integrated circuit according to claim 2, wherein said first unbuffer signal is initiated by the slave and said second unbuffer signal is initiated by the master.”

136. Claim 2 is obvious as discussed above. *See* IX.A.9, above. In my opinion, Rădulescu also renders obvious the additional limitation of claim 4 because it teaches that its unbuffer signals are initiated by the IP modules in the symmetrical NoC. *See* IX.A.9.

11. Claim 6: “The integrated circuit according to claim 2, wherein the master is adapted to initiate said first unbuffer signal and said second unbuffer signal.”

137. Claim 2 is obvious as I discussed above. *See* IX.A.9, above. In my opinion, Rădulescu also renders obvious the additional limitation of claim 6 because a POSITA would have understood based on Rădulescu’s disclosure that the master IP module initiates the flush signals (the claimed the first and second unbuffer signals), as I discussed above with respect to claim 18. *See* IX.A.3, above.

12. Claim 8: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using messages of a message-passing communication scheme, wherein the messages contain a message header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said message header, respectively, even if less than said first amount and said second amount of data is buffered in said first wrapper and said second wrapper.”

138. Claim 1 is obvious as discussed above. *See* IX.A.8, above. In my opinion, Rădulescu also renders obvious the additional limitation of claim 8. As Rădulescu explains, communication over its NoC “is performed using a transaction-based protocol, where master IP modules issue *request messages* ... that are executed by the addressed slave modules, which may respond with a *response message*” Ex. 1005 (Rădulescu) at 1; *see also* Ex. 1005 (Rădulescu) at 2, fig. 9. A POSITA would have understood that such network communication is a message-passing communication scheme. Rădulescu further discloses that the messages in

such scheme contain a message header. *See* Ex. 1005 (Rădulescu) at fig. 7, 4 (referring to “message header” of a flush signal from IP module), 3 (referring to “header of the packets of the data” transmitted over NoC).

139. A POSITA also would have understood that the first and second wrappers are adapted to transfer buffered data in response to information in the message header even if the amount of buffered data is below the threshold first and second amounts because Rădulescu’s flush signal causes this to occur, as I explained with respect to claim 2, and is associated with “a bit in the message header” that causes the threshold to be “temporarily overrid[den],” as I explained with respect to claim 21. Ex. 1005 (Rădulescu) at 4.

- 13. Claim 9: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using packets of a packet based communication scheme, wherein the packets contain a packet header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said packet header, respectively, even if less than said first amount and said second amount of data is buffered in said first and second wrapper.”**

140. Claim 1 is obvious as discussed above. *See* IX.A.8, above. In my opinion, Rădulescu also renders obvious the limitation of claim 9. Rădulescu’s master and slave IP modules communicate through request and response messages, which “are transported by means of *packets*.” Ex. 1005 at 2; *see also* Ex. 1005 (Rădulescu) at 3 (referring to “packet-based communication of the NoC” and

“packetiz[ing] messages”), 4 (referring to “number of bits in the packet header”). A POSITA would have understood that such network communication is a packet-based communication scheme. As Rădulescu also explains, its data packets comprise “[a] packet header.” Ex. 1005 (Rădulescu) at 3. In addition, as I described above with respect to claim 2, Rădulescu discloses a “flush signal [that] forces credits to be sent even when they are below their threshold,” where the “credits are piggybacked on packets” and are limited to “the given number of bits in the packet header.” Ex. 1005 (Rădulescu) at 4. Such credits indicate available space in the second data in the second wrapper (at the slave-side of the interconnect). As such, when the second flush signal forces credits to be sent even when they are below their threshold, the second data is flushed from the second wrapper to be transferred across the interconnect. A POSITA would have therefore understood that the first and second wrappers are adapted to transfer buffered data in response to information in the packet header even if the amount of buffered data is below the threshold first and second amounts. Ex. 1005 at 3–4; *see* IX.A.12, above.

14. Claim 11

141. In my opinion, Rădulescu in view of Bahadiroglu teaches all the limitations of claim 11 and thus would have rendered the claim obvious.

a. Claim 11[pre]: “A data processing system, comprising:”

142. I was not asked to analyze whether the preamble of Claim 11 is limiting. In any event, in my opinion, Rădulescu teaches a data processing system. For example, Rădulescu discloses processing modules of an SoC that execute commands and perform computation. *E.g.*, Ex. 1005 (Rădulescu) at Abstract, 1 (“master IP modules issue request messages . . . that are executed by the addressed slave modules, which may respond with a response message”), 2 (“the communication services of the Æthereal NoC are defined to meet the following goals: (a) decouple *computation (IP modules)* from communication (NoC)...”), 3-5 (describing “video pixel processing”). Rădulescu’s system is therefore a data processing system.

b. Claim 11[a]: “a plurality of processing modules each having a memory and being connected with an interconnect, wherein a first processing module communicates to a second processing module using transactions, wherein the first processing module is a master for providing requests and the second processing module is a slave for providing responses to the requests; and”

143. *See* IX.A.8.b, above.

- c. **Claim 11[b]: “a first interface unit coupled to the slave and including a first wrapper for buffering data from the slave to be transferred over the interconnect, the first interface unit further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first wrapper, and wherein the buffered data are transferred when said first optimal amount of data has been buffered by the first wrapper, and wherein the first interface unit including the first wrapper and the first determination unit is connected between the slave and the interconnect; and”**

144. *See* IX.A.8.c, above.

- d. **Claim 11[c]: “a second interface unit coupled to the master and including a second wrapper for buffering data from the master to be transferred over the interconnect, the second interface unit further including a second determination unit, wherein second first determination unit is configured to determine a second optimal amount of data to be buffered by the second wrapper, and wherein the buffered data are transferred when said second optimal amount of data has been buffered by the second wrapper, and wherein the second interface unit including the second wrapper and the second determination unit is connected between the master and the interconnect,”**

145. *See* IX.A.8.d, above.

- e. **Claim 11[d]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

146. *See* IX.A.8.e, above.

B. Ground 2: Claims 1, 2, 4, 6, 8–12, 13, and 18–21 are Rendered Obvious by Rădulescu in view of Petersen

147. In my opinion, claims 1, 2, 4, 6, 8–12, 13, and 18–21 are rendered obvious by the combination of Rădulescu and Petersen.

148. As a threshold matter, a POSITA would have been motivated to combine the teachings of Rădulescu with those of Petersen in order to create a NOC with improved performance and efficiency.

149. As discussed above, Rădulescu discloses “optimiz[ing] the NoC utilization” using a “configurable threshold mechanism” to determine when the “amount of sendable data” queued in the network interface is sufficiently large. *See* IX.A, above. By waiting to send data until the threshold is met, this mechanism

produces longer packets that use the NoC more efficiently and consume less bandwidth to send a given amount of data. Ex. 1005 (Rădulescu) at 4. A POSITA would have understood that different, known ways of implementing Rădulescu's configurable threshold existed at the time of the alleged invention. For example, the configurable threshold could be set with a fixed value (*e.g.*, a value programmed into a register), or it could be set with an adjustable value (*e.g.*, a value that changes at runtime according to conditions in the NoC). Ex. 1030 (Heinlein 1994) at 47 n.6 (teaching a mechanism for “dynamically adjust[ing]” an amount of stored data to send). Rădulescu does not select a specific implementation for its configurable threshold mechanism.

150. Petersen is directed to the same goal of optimizing network utilization, discloses a specific way to implement a threshold mechanism like the one disclosed by Rădulescu, and teaches various benefits of using Petersen's specific threshold mechanism. A POSITA would have therefore been motivated to configure Rădulescu configurable threshold mechanism using Petersen's teachings related to a runtime-adjustable threshold mechanism. In particular, Petersen discloses a “dynamically programmable” threshold as a specific implementation of setting a configurable threshold like that described by Rădulescu and scheduling packet transmissions. Ex. 1007 (Petersen) at 4:40-60. Petersen explains that its configurable “threshold” is used to determine whether there is a sufficient amount of data buffered

in a network interface before transmitting that data in order to “optimize performance,” *see, e.g.*, Ex. 1007 (Petersen) at Abstract, which is consistent with and nearly identical to Rădulescu’s disclosure of using a configurable threshold to “skip[] a channel as long as the sendable data is below the threshold,” such that data is not transmitted until there is a sufficient amount of data buffered. Ex. 1005 (Rădulescu) at 4. Petersen further describes that its threshold mechanism is dynamically programmed based on network conditions, such as “system latencies or bandwidth limitations,” to avoid “underrun” conditions and “unnecessary delays.” Ex. 1007 (Petersen) at 4:40–60, 29:39–46.

151. A POSITA would have understood that configuring Rădulescu’s threshold mechanism using Petersen’s “dynamically programmable” threshold mechanism would improve the performance and efficiency of Rădulescu’s NoC by optimizing both (1) the “amount of data” to be transferred and the (2) timing of transmission (for example, to avoid unnecessary delays) based on real-time network conditions, which together accomplish Rădulescu’s stated goal of “optimiz[ing] the NoC utilization.” Ex. 1007 (Petersen) at 2:18–34, 4:68–5:3, 29:39–57; Ex. 1005 (Rădulescu) at 4. Specifically, a POSITA would have recognized that using Petersen’s threshold mechanism results in the transmission of larger data packets when network conditions allow, which in turn reduces the number of packets and packet headers that are ultimately transmitted (for a given data amount), therefore

allowing more data to travel over the NoC more efficiently. Using Petersen's dynamically programmable threshold would also tailor the packet size to the network conditions, which prevents the downfalls associated with transmitting packets that are too large for the current network conditions, such as buffers overflowing and packets being dropped. When packets are dropped, the data must be retransmitted, which is wasteful and inefficient. Similarly, using Petersen's mechanism maximizes usage of the network bandwidth and manages traffic by optimizing the timing of data transmission, which likewise increases the likelihood of successful transmissions and avoids wasteful retransmissions. Moreover, a POSITA also would have been motivated to use Petersen's threshold mechanism based on Petersen's disclosure of the benefits of using a transmit buffer system, "while maintaining the communications throughput available from the simpler FIFO based systems." Compare Ex. 1005 (Rădulescu) at 3, 5 (disclosing use of FIFO buffers), with Ex. 1007 (Petersen) at 1:47–63, 2:7–10 (balancing benefits of transmit and FIFO buffers). A POSITA would have further recognized Petersen discloses striving to decouple the host from communication, which is a goal that Rădulescu shares. *See* Ex. 1007 (Petersen) at 1:65–67 ("[T]he prior art systems which use transmit data buffers require the host or sending system to manage the transmit data buffer."); Ex. 1005 (Rădulescu) at 1-2. For all of these reasons, a skilled artisan

would have been motivated to combine Rădulescu's and Petersen's teachings to arrive at the claimed invention.

152. A POSITA also would have had a reasonable expectation of success in making such combination. A POSITA would have understood that combining Rădulescu and Petersen merely combines prior art elements (configuring Rădulescu's threshold mechanism with the functionality of Petersen's threshold mechanism) according to known methods (implementing Petersen's detailed disclosure of a specific type of configurable threshold mechanism) to yield predictable results (network optimization). Moreover, both Rădulescu and Petersen occupy the same field of optimizing data communication in a network. Ex. 1005 (Rădulescu) at 1, 4; Ex. 1007 (Petersen) at Abstract. A POSITA would have reasonably expected that use of Petersen's known technique for optimizing data communication in a network through its dynamically programmable threshold would improve Rădulescu's NoC in the same way. It would require nothing more than routine implementation details to combine Petersen's mechanism with Rădulescu's NoC, and Petersen provides an extensive disclosure of such implementation details throughout its specification. *See, e.g.*, Ex. 1007 (Petersen) at 2:18–34, 4:68–5:3, 29:39–57. A POSITA would also have understood that Petersen's mechanism is directed to improving performance in a network, which likewise applies to other networks that can benefit from such performance improvements, such as on-chip

networks. For example, Petersen's dynamically programmable threshold mechanism would optimize rather than exhaust the resources in Rădulescu's NoC while simultaneously maximizing bandwidth, capacity, and traffic management efficiencies through Rădulescu's NoC at minimal resource cost. Ex. 1007 (Petersen) at 2:18–34, 4:68–5:3, 29:39–57.

153. Furthermore, as I explained in the technology background section above, NoCs have borrowed many design features from off-chip interconnection networks. For example, as described in one text devoted to networks-on-chip “[d]esigning a network for on-chip communication is similar to the problem of designing a local area computer network with some computing and communication requirements.” Ex. 1009 (Jantsch) at 56 (native page 102); *see also* Ex. 1015 (Ye) at Abstract (“On-chip networks borrow features and design methods from those used in parallel computing clusters and computer system area networks”). Another chapter in the same text explained that addressing issues related to the optimization of network usage—as Rădulescu and Petersen do—is substantially similar in on-chip and off-chip networks: “The resource allocation, optimization of network usage and verification of performance and correctness are the main problems that are basically similar to what distributed and parallel system designers have to face.” Ex. 1009 (Jantsch) at 21 (native page 33). Indeed, the '800 patent itself acknowledges that many of the issues faced by networks on chip had been studied in the context of

off-chip networks and that results of those studies are applicable to on-chip networks as well. Ex. 1001 ('800 patent) at 1:48–64.

1. Claim 12

154. In my opinion, Rădulescu in view of Petersen teaches all the limitations of claim 12, and thus would have rendered that claim obvious.

a. Claim 12[preamble]: “An integrated circuit comprising:”

155. *See* IX.A.1.a, above.

b. Claim 12[a]: “a plurality of processing modules including a first processing module having a first processing memory and a second processing module having a second processing memory”

156. *See* IX.A.1.b, above.

c. Claim 12[b]: “a plurality of interconnect modules including a first interconnect module and a second interconnect module, wherein the first processing module is connected to the first interconnect module and the second processing module is connected to the second interconnect module”

157. *See* IX.A.1.c, above.

- d. **Claim 12[c]: “a network for connecting the first interconnect module to the second interconnect module in order to provide a communication connection between the first processing module and the second processing module, wherein the first interconnect module is connected between the first processing module and the network, and wherein the second interconnect module is connected between the second processing module and the network”**

158. *See* IX.A.1.d, above.

- e. **Claim 12[d]: “wherein the first interconnect module includes a first memory for buffering first data from the first processing module, the first interconnect module further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first memory, and the second interconnect module includes a second memory for buffering second data from the second processing module, the second interconnect module further including a second determination unit, wherein the second determination unit is configured to determine a second optimal amount of data to be buffered by the second memory”**

159. In my opinion, Rădulescu in view of Petersen renders this limitation obvious. As I discussed in Ground 1, Rădulescu discloses that its network interfaces NI2 and NI1 (the claimed first and second interconnect modules) include source queues (the claimed first and second memories) that buffer sendable data received from their associated IP blocks (the claimed first and second processing modules). *See* IX.A.1.e, above. Rădulescu also discloses that each of its network interfaces includes a scheduler with “a configurable threshold mechanism” that prevents

buffered data from being transmitted “as long as the sendable data is below the threshold,” in order to “optimize the NoC utilization.” *See* Section IX.A.1.e above.

160. A POSITA also would have found this element obvious in view of Petersen, which discloses a determination unit configured to determine an optimal amount of data to be buffered in a network interface memory.

161. In particular, Petersen discloses a network interface system for transmitting data from a “transmit buffer” only after a “threshold determination indicates that a sufficient portion of the [data] frame is resident in the transmit buffer.” Ex. 1007 (Petersen) at 2:14–27, *see also* Ex. 1007 (Petersen) at 4:13–17 (“The early transmit logic 6A makes a threshold determination for a given frame being transferred from the host 1 into the adapter memory 9. When the threshold is met, then the adapter 6 begins transmitting the frame across the network.”), Abstract, 4:40–45, 4:64–5:5, 5:56–61, 31:16–28, figs. 1–2. Petersen explains that this “threshold determination” is based on comparing “the amount of data of a frame which is downloaded by the host to the transmit buffer” to a “threshold value” housed in a “threshold store,” wherein the threshold value “may be set by the host system to *optimize* performance” Ex. 1007 (Petersen) at 2:31–34, 4:40–45; *see also* Ex. 1007 (Petersen) at 4:46–47 (“The threshold store 43, in a preferred system, is dynamically programmable by the host computer 30.”), Abstract, 2:27–31, 5:5–13, 21:14–16, 21:31–34, 23:20–27, 24:48–52, 24:59–64, 25:3–18, 29:39–57, 31:16–

28, FIG. 12. For example, Petersen states that “status information” supplied to the host system that “includes indications of underrun conditions” allows “the host to optimize the value” of the threshold. Ex. 1007 (Petersen) at 4:56–60; *see also* Ex. 1007 (Petersen) at 29:39–57, 31:34–37, 32:48–55.

162. A POSITA would have understood that Petersen’s optimal “threshold value” represents an optimal amount of data to be buffered, because the threshold determines the amount of data that must be buffered in the network interface memory before the data can be transmitted across the network. As Petersen explains, “packets or frames of data” are transmitted “across a communications network” only after “the amount of data” in the transmit buffer satisfies the “threshold value,” which constitutes the optimal amount of data. Ex. 1007 (Petersen) at 1:17–21, 2:18–27; *see also* 4:40–45 (stating the “threshold value ... indicates an amount of data of a frame that must be resident in the frame buffer 34 before transmission of that frame may be initiated”), 4:68–5:3 (“When the threshold amount of data is resident in the buffer 34, then the transmit logic 39 is instructed to begin transmission of the frame.”), Abstract. Moreover, given Petersen’s technique of optimizing the threshold value based on current conditions in the system, Ex. 1007 (Petersen) at 4:56–60; *see also* Ex. 1007 (Petersen) at 29:39–57, 31:34–37, 32:48–55, a POSITA would have understood that Petersen teaches determination of both first and second optimal amounts of data as conditions change.

163. For the reasons discussed above, a POSITA would have looked to Petersen's disclosures regarding a dynamically programmable threshold to configure Rădulescu's threshold, and would have had a reasonable expectation of success in doing so. *See* IX.B above.

- f. **Claim 12[e]: “the first interconnect module transferring the first data to the second processing module when the first data buffered in the first memory reaches the first optimal amount, and the second interconnect module transferring the second data to the first processing module when the second data buffered in the second memory reaches the second optimal amount”**

164. In my opinion, Rădulescu in view of Petersen renders this limitation obvious. As discussed above with respect to claim element 12[d], Rădulescu describes transmitting data buffered in the source queues (the claimed first and second memories) of each of its network interfaces NI2 and NI1 (the claimed the first and second interconnect modules) to the destination queues of the other network interface, and then to a target IP block (from the first processing module to the second and from the second processing module to the first, as claimed), when the amount of buffered data is above “a configurable threshold.” Ex. 1005 (Rădulescu) at 4; *see also* Ex. 1005 (Rădulescu) at fig. 8 (showing data flows through Rădulescu's NoC from slave A to master B, and vice versa). As further discussed, Petersen discloses transmitting data over a network when “the threshold amount of data is resident in the buffer.” *See* Ex. 1007 (Petersen) at 2:18–34, 4:68–5:3; *see also*

Ex. 1007 (Petersen) at at Abstract, 2:52–53, 4:11–17, 4:30–45, 4:56–60, 5:3–13, 21:14–16, 21:31–34, 24:59–25:18, 29:39–40, 31:16–33, figs. 1–2, 12.

- g. Claim 12[f]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave”**

165. In my opinion, Rădulescu discloses this limitation. *See* IX.A.1.g, above. Petersen also describes determining an optimal threshold value, which dictates when buffered data is transmitted (the claimed optimal moment), based on communication properties such as “system latencies or bandwidth limitations,” to avoid “underrun” conditions and “unnecessary delays.” Ex. 1007 (Petersen) at 4:40–60, 29:39–46. A POSITA would understand that by taking these conditions into account and setting the threshold value to, for example, avoid delays, Petersen’s mechanism is configured to determine an optimal moment for sending the buffered data. In addition, latency and bandwidth (*i.e.*, throughput) are both among the claimed communication properties.

166. For the reasons discussed above, a POSITA would have looked to Petersen’s disclosures regarding dynamically programmable threshold to configure Rădulescu’s threshold, and would have had a reasonable expectation of success in doing so. *See* IX.B, above.

- h. Claim 12[g]: “wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery”**

167. In my opinion, Rădulescu discloses this limitation and Rădulescu in view of Petersen renders this limitation obvious. As discussed above, Rădulescu discloses that many of the listed communication properties are used in the operation of its NoC. *See* IX.A.1.h, above. Moreover, a POSITA would have understood that Rădulescu’s scheduler considers these communication properties in determining when to transmit data buffered in the source queues of Rădulescu’s NI1 and NI2. Ex. 1005 (Rădulescu) at 3-4. Petersen likewise describes determining when to transmit data based on the listed communication properties, including latency and throughput. *See* IX.B.1.g, above.

168. Rădulescu and Petersen—each independently and in combination—disclose determining the moment of transmission based on one or more of the claimed “communication properties.” As explained in Ground I, to the extent that any of the claimed communication properties are not explicitly disclosed by Rădulescu or Petersen, such properties were well-known to a POSITA, and it would

have been obvious to consider such properties in scheduling transmissions. *See* IX.A.1.h, above.

2. **Claim 13: “The integrated circuit of claim 12, wherein the plurality of interconnect modules manages the communication connection freeing the plurality of processing modules to perform data processing.”**

169. In my opinion, Claim 12 is obvious as I discussed above. *See* Section IX.B.1. In addition, in my opinion, Rădulescu discloses the limitation of claim 13 for the reasons described in Ground 1. *See* Section IX.A.2.

3. **Claim 18: “The integrated circuit of claim 12, wherein the first processing module provides a first flushing command to flush the first memory and a second flushing command to flush the second memory so that the first data and the second data are sent without waiting to reach the first amount and the second amount.”**

170. *See* IX.A.1, IX.A.3, above.

4. **Claim 19: “The integrated circuit of claim 18, wherein the first processing module provides the first flushing command and the second flushing command at least one of independently and simultaneously.”**

171. *See* IX.A.3, IX.A.4, above.

5. **Claim 20: “The integrated circuit of claim 18, wherein the second flushing command includes an indication to flush the second memory so that the first interconnect module forwards the second flushing command to the second interconnect module for flushing the second data from the second memory.”**

172. *See* IX.A.3, IX.A.5, above.

6. **Claim 21:** “The integrated circuit of claim 12, wherein the first processing module includes flush information in a header of a message to cause all messages from a same connection that have been sent earlier to be flushed from at least one of the first memory and the second memory.”

173. *See* IX.B.1, IX.A.6, above.

7. **Claim 10**

174. In my opinion, Rădulescu in view of Petersen teaches all the limitations of claim 10, and thus would have rendered that claim obvious.

- a. **Claim 10[pre]:** “A method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions”

175. *See* IX.A.7.a, above.

- b. **Claim 10[a]:** “configuring the first processing module having a first memory as a master the provides requests”

176. *See* IX.A.7.b, above.

- c. **Claim 10[b]:** “configuring the second processing module having a second memory as a slave the provides responses to the requests”

177. *See* IX.A.7.c, above.

- d. **Claim 10[c]:** “connecting the master to a master interface unit of the interface units”

178. *See* IX.A.7.d, above.

- e. **Claim 10[d]: “connecting the master interface unit to the interconnect so that the master interface unit is between the master and the interconnect”**

179. *See* IX.A.7.e, above.

- f. **Claim 10[e]: “connecting the slave to a slave interface unit of the interface units”**

180. *See* IX.A.7.f, above.

- g. **Claim 10[f]: “connecting the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect”**

181. *See* IX.A.7.g, above.

- h. **Claim 10[g]: “determining by a master determination unit of the master interface unit a first optimal amount of data to be buffered by a master wrapper of the master interface unit”**

182. *See* § IX.B.1.e, IX.A.7.h, above.

- i. **Claim 10[h]: “determining by a slave determination unit of the slave interface unit a second optimal amount of data to be buffered by a slave wrapper of the slave interface unit”**

183. *See* § IX.B.1.e, IX.A.7.i, above.

- j. **Claim 10[i]: “buffering by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered”**

184. *See* IX.B.1.e, above.

- k. Claim 10[j]: “transferring the buffered data from the slave wrapper to the master wrapper when said first optimal amount of data has been buffered by the slave wrapper”**

185. *See* IX.B.1.f, above.

- l. Claim 10[k]: “buffering by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper”**

186. *See* IX.B.1.e, above.

- m. Claim 10[l]: “transferring the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper,”**

187. *See* IX.B.1.f, above.

- n. Claim 10[m]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave,”**

188. *See* IX.B.1.g, above.

- o. Claim 10[n]: “wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

189. *See* IX.B.1.h, above.

8. Claim 1

190. In my opinion, Rădulescu in view of Petersen teaches all the limitations of claim 1 and thus would have rendered the claim obvious.

- a. Claim 1[pre]: “An integrated circuit comprising:”**

191. *See* IX.B.1.a, above.

- b. Claim 1[a]: “a plurality of processing modules being connected with an interconnect and each of the plurality of processing modules having a memory, wherein a first processing module communicates to a second processing module using transactions, wherein the first processing module is a master for providing requests and the second processing module is a slave for providing responses to the requests;”**

192. *See* IX.B.7.a–IX.B.7.c, above.

- c. **Claim 1[b]: “a first interface unit coupled to the slave and including a first wrapper for buffering data from the slave to be transferred over the interconnect to the master, the first interface unit further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first wrapper, and wherein the buffered data are transferred when said first optimal amount of data has been buffered by the first wrapper, and wherein the first interface unit including the first wrapper and the first determination unit is connected between the slave and the interconnect; and”**

193. *See* IX.B.7.f–IX.B.7.g, IX.B.7.i–IX.B.7.k, above.

- d. **Claim 1[c]: “a second interface unit coupled to the master and including a second wrapper for buffering data from the master to be transferred over the interconnect, the second interface unit further including a second determination unit, wherein second determination unit is configured to determine a second optimal amount of data to be buffered by the second wrapper, and wherein the buffered data are transferred when said second optimal amount of data has been buffered by the second wrapper, and wherein the second interface unit including the second wrapper and the second determination unit is connected between the master and the interconnect,”**

194. *See* IX.B.7.d–IX.B.7.e, IX.B.7.h, IX.B.7.l–IX.B.7.m, above.

- e. **Claim 1[d]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

195. *See* IX.B.7.n–IX.B.7.o, above.

- 9. **Claim 2: “The integrated circuit according to claim 1, wherein the first interface unit includes a first unbuffer-flag unit for providing a first unbuffer signal and the second interface unit includes a second unbuffer-flag unit for providing a second unbuffer signal, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to the first unbuffer signal and the second unbuffer signal, respectively, even if less than said first optimal amount and said second optimal amount of data is buffered in said first wrapper and said second wrapper.”**

196. *See* IX.B.8, IX.A.9, above.

- 10. **Claim 4: “The integrated circuit according to claim 2, wherein said first unbuffer signal is initiated by the slave and said second unbuffer signal is initiated by the master.”**

197. *See* IX.B.9, IX.A.10, above.

11. **Claim 6: “The integrated circuit according to claim 2, wherein the master is adapted to initiate said first unbuffer signal and said second unbuffer signal.”**

198. *See* IX.B.9, IX.A.10, above.

12. **Claim 8: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using messages of a message-passing communication scheme, wherein the messages contain a message header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said message header, respectively, even if less than said first amount and said second amount of data is buffered in said first wrapper and said second wrapper.”**

199. *See* IX.B.8, IX.A.12, above.

13. **Claim 9: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using packets of a packet based communication scheme, wherein the packets contain a packet header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said packet header, respectively, even if less than said first amount and said second amount of data is buffered in said first and second wrapper.”**

200. *See* IX.B.8, IX.A.13, above.

14. Claim 11

201. In my opinion, Rădulescu in view of Petersen teaches all the limitations of claim 11 and thus would have rendered the claim obvious.

- a. **Claim 11[pre]: “A data processing system, comprising:”**

202. *See* IX.A.14.a, above.

- b. Claim 11[a]: “a plurality of processing modules each having a memory and being connected with an interconnect, wherein a first processing module communicates to a second processing module using transactions, wherein the first processing module is a master for providing requests and the second processing module is a slave for providing responses to the requests; and”**

203. *See* IX.B.8.b, above.

- c. Claim 11[b]: “a first interface unit coupled to the slave and including a first wrapper for buffering data from the slave to be transferred over the interconnect, the first interface unit further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first wrapper, and wherein the buffered data are transferred when said first optimal amount of data has been buffered by the first wrapper, and wherein the first interface unit including the first wrapper and the first determination unit is connected between the slave and the interconnect; and”**

204. *See* IX.B.8.c, above.

- d. **Claim 11[c]: “a second interface unit coupled to the master and including a second wrapper for buffering data from the master to be transferred over the interconnect, the second interface unit further including a second determination unit, wherein second first determination unit is configured to determine a second optimal amount of data to be buffered by the second wrapper, and wherein the buffered data are transferred when said second optimal amount of data has been buffered by the second wrapper, and wherein the second interface unit including the second wrapper and the second determination unit is connected between the master and the interconnect,”**

205. *See* IX.B.8.d, above.

- e. **Claim 11[d]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

206. *See* IX.B.8.e, above.

C. Ground 3: Claims 1, 2, 4, 6, 8–12, 13, and 18–21 are Rendered Obvious by Rădulescu in view of Bahadiroglu and Rădulescu-WO

207. In my opinion, claims 1, 2, 4, 6, 8–12, 13, and 18–21 are rendered obvious by the combination of Rădulescu, Bahadiroglu, and Rădulescu-WO.

208. As a threshold matter, a POSITA would have been motivated to combine Rădulescu and Bahadiroglu for the reasons described in Ground 1. *See* IX.A, above. A POSITA also would have been motivated to combine Rădulescu and Rădulescu-WO. First, Rădulescu and Rădulescu-WO were both authored by Andrei Rădulescu and Kees Goossens around the same time period. *Compare* Ex. 1008 (Rădulescu-WO) at Cover *with* Ex. 1005 (Rădulescu) at 1. Second, Rădulescu and Rădulescu-WO are directed to the very same network-on-chip technology. Both references disclose an integrated circuit, and more specifically “a plurality of processing modules and a network arranged for providing connections between processing modules and a method for exchanging messages in such an integrated circuit.” Ex. 1008 (Rădulescu-WO) at 1:1-4; Ex. 1005 (Rădulescu) at 1-4, FIG. 8. Both references’ NoC also specifically supports data communication between “processing modules,” over “connections,” which each have “a set of connection properties.” Ex. 1008 (Rădulescu-WO) at Abstract, 9:5–15, 9:18–21, 11:5–11, 11:17–22; Ex. 1005 (Rădulescu) at 1 (“We provide transport-layer services by defining connections (e.g., point-to-point or multicast) configured for specific properties (e.g., throughput, ordering).”), FIG. 8. Third, a POSITA considering

Rădulescu and its goal of “optimiz[ing] the NoC utilization” would have looked for similar network configurations that used known communication properties to optimize communications over an NoC, and such POSITA would have found Rădulescu-WO’s disclosure of using such communication properties for that specific purpose, further motivating the POSITA to combine these references. Fourth, Rădulescu-WO discloses additional implementation details pertaining to certain features of Rădulescu’s NoC and a POSITA would have been motivated to consider the two references together to have the benefit of these additional details. For example, Rădulescu-WO, like Rădulescu, discloses an NoC with a plurality of “processing modules,” such as master and slave modules, that are connected through “network interface means NI, associated to each of said [processing] modules.” Ex. 1008 (Rădulescu-WO) at 9:10-10:20. But Rădulescu-WO teaches that each of the connections between these modules “comprises a set of communication channels each having a set of connection properties,” such as ordering, flow control, throughput, and lossiness (among others), and that such connection properties “can be adjusted independently” to maximize “effective usage of the properties of the network.” Ex. 1008 (Rădulescu-WO) at 9:5-17, 11:17-22. For all of these reasons, a POSITA would have been motivated to combine these references and would have had a reasonable expectation of success, particularly given that the references pertain to the same NoC and have the same authors.

1. Claim 12

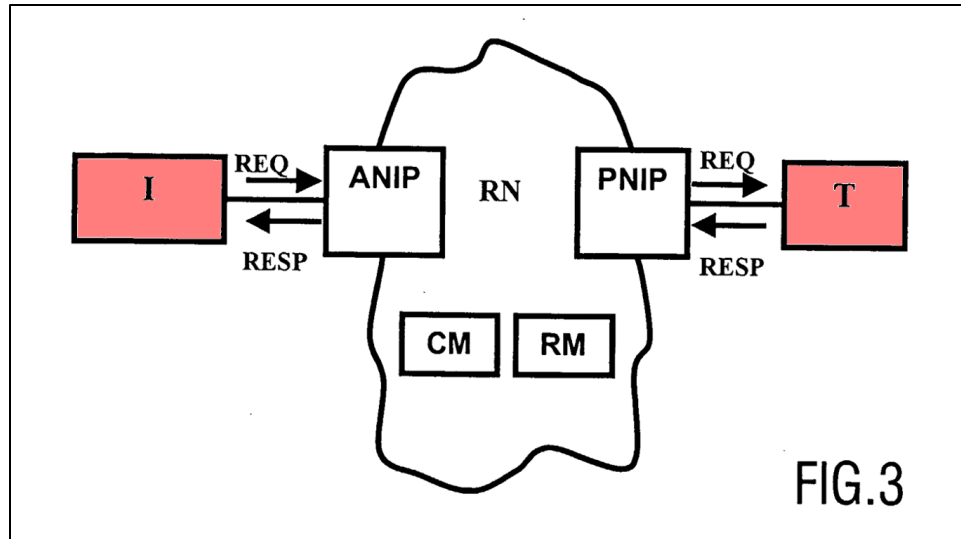
209. In my opinion, Rădulescu in view of Bahadiroglu and Rădulescu-WO teaches all the limitations of claim 12, and thus would have rendered that claim obvious.

a. Claim 12[preamble]: “An integrated circuit comprising”

210. *See* IX.A.1.a, above.

b. Claim 12[a]: “a plurality of processing modules including a first processing module having a first processing memory and a second processing module having a second processing memory”

211. As discussed in Ground 1, Rădulescu discloses this limitation and/or renders it obvious. *See* IX.A.1.b, above. In addition, processing modules having processing memory would have been obvious in view of Rădulescu-WO. Rădulescu-WO discloses “a plurality of processing modules”—also described as “intellectual property blocks IPs”—including “masters” and “slaves.” Ex. 1008 (Rădulescu-WO) at Abstract, 1:1, 2:7, 12:13–15; *see also* Ex. 1008 (Rădulescu-WO) at 9:10–12, 11:5–6, 11:23–24, 11:31–33, 13:1–2, 13:6, 14:28–31, 25:1–9. Two such processing modules—“initiator module I” and “target module T”—are shown in red below.



Ex. 1008 (Rădulescu-WO) at fig. 3 (annotated); *see also* Ex. 1008 (Rădulescu-WO) at Abstract, 12:5–1, figs. 1–2.

212. A POSITA would have understood that Rădulescu-WO’s processing modules, including the masters and slaves, are processing modules having processing memory. Rădulescu-WO states “[t]he modules ... can be so-called intellectual property blocks IPs (computation elements, memories or a subsystem which may internally contain interconnect modules).” Ex. 1008 (Rădulescu-WO) at 12:13–15; *see also* Ex. 1008 (Rădulescu-WO) at 6:7–9, 22:23–23:1. This disclosure is identical to the ’800 Patent’s disclosure of processing modules: “[t]he modules ... can be so-called intellectual property blocks IPs (computation elements, memories or a subsystem which may internally contain interconnect modules).” Ex. 1001 at 5:51–55, 8:48–51, 10:67–11:3; *see also* § IX.A.1.b, above.

213. In addition, Rădulescu-WO describes that the modules have within them “a memory, or a register.” Ex. 1008 (Rădulescu-WO) at 6:7–9. Rădulescu-WO also teaches that “NoCs also decouple computation from communication.” Ex. 1008 (Rădulescu-WO) at 1:25. A POSITA would understand that to achieve this functionality, memory must exist in order to hold the data that is intended for computation.

- c. **Claim 12[b]: “a plurality of interconnect modules including a first interconnect module and a second interconnect module, wherein the first processing module is connected to the first interconnect module and the second processing module is connected to the second interconnect module”**

214. *See* IX.A.1.c, above.

- d. **Claim 12[c]: “a network for connecting the first interconnect module to the second interconnect module in order to provide a communication connection between the first processing module and the second processing module, wherein the first interconnect module is connected between the first processing module and the network, and wherein the second interconnect module is connected between the second processing module and the network”**

215. *See* IX.A.1.d, above.

- e. **Claim 12[d]: “wherein the first interconnect module includes a first memory for buffering first data from the first processing module, the first interconnect module further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first memory, and the second interconnect module includes a second memory for buffering second data from the second processing module, the second interconnect module further including a second determination unit, wherein the second determination unit is configured to determine a second optimal amount of data to be buffered by the second memory”**

216. *See* IX.A.1.e, above.

- f. **Claim 12[e]: “the first interconnect module transferring the first data to the second processing module when the first data buffered in the first memory reaches the first optimal amount, and the second interconnect module transferring the second data to the first processing module when the second data buffered in the second memory reaches the second optimal amount”**

217. *See* IX.A.1.f, above.

- g. **Claim 12[f]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave”**

218. *See* IX.A.1.g, above.

- h. Claim 12[g]: “wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery”**

219. As discussed in Ground 1, Rădulescu in view of Bahadiroglu renders this limitation obvious. *See* IX.A.1.h, above. To the extent PO argues that the full list of claimed communication properties is not disclosed or rendered obvious by Rădulescu in view of Bahadiroglu, however, it is obvious in view of Rădulescu-WO. Rădulescu-WO discloses all the listed communication properties and their use in the operation of its NoC. Ex. 1008 (Rădulescu-WO) at Abstract, 9:5–7, 9:18–23, 11:5–22, 16:32–34. It explains that “communication between the modules are performed over connections” that are made up of “channels” having “connection properties,” which can be configured to maximize efficient use of NoC resources. Ex. 1008 (Rădulescu-WO) at 11:8–11; *see also* Ex. 1008 (Rădulescu-WO) at Abstract, 9:5–7, 9:18–23, 12:19–22, 14:9–13, 15:17–18, 16:32–34. Rădulescu-WO states:

“[t]he connection properties may include ordering (data transport in order), flow control (a remote buffer is reserved for a connection, and a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data), throughput (a lower bound on throughput is guaranteed), latency (upper bound for latency is guaranteed), the lossiness (dropping of data), transmission

termination, transaction completion, data correctness, priority, or data delivery.”

Ex. 1008 (Rădulescu-WO) at 11:17–22; *see also* Ex. 1008 (Rădulescu-WO) at 16:32–34. This is exactly the same list of connection properties disclosed and claimed in the ’800 Patent. *See* Ex. 1001 at 5:27–35, claim 12. Rădulescu-WO describes many of these properties and their operation in detail. *See* Ex. 1008 (Rădulescu-WO) at 3:9–4:22, 5:21–9:2.

220. As explained above in Ground 1, a POSITA would have found it obvious to consider the connection properties listed in Rădulescu-WO in determining the “optimum inter-packet interval” to further optimize data communication with a reasonable expectation of success. IX.A.1.h, above.

2. **Claim 13: “The integrated circuit of claim 12, wherein the plurality of interconnect modules manages the communication connection freeing the plurality of processing modules to perform data processing.”**

221. *See* IX.C.1, IX.A.2, above.

3. **Claim 18: “The integrated circuit of claim 12, wherein the first processing module provides a first flushing command to flush the first memory and a second flushing command to flush the second memory so that the first data and the second data are sent without waiting to reach the first amount and the second amount.”**

222. *See* IX.C.1, IX.A.3, above.

4. **Claim 19:** “The integrated circuit of claim 18, wherein the first processing module provides the first flushing command and the second flushing command at least one of independently and simultaneously.”

223. See IX.C.4, IX.A.4, above.

5. **Claim 20:** “The integrated circuit of claim 18, wherein the second flushing command includes an indication to flush the second memory so that the first interconnect module forwards the second flushing command to the second interconnect module for flushing the second data from the second memory.”

224. See IX.C.4, IX.A.5, above.

6. **Claim 21:** “The integrated circuit of claim 12, wherein the first processing module includes flush information in a header of a message to cause all messages from a same connection that have been sent earlier to be flushed from at least one of the first memory and the second memory.”

225. See IX.C.1, IX.A.6, above.

7. **Claim 10**

226. In my opinion, Rădulescu-WO in view of Bahadiroglu teaches all the limitations of claim 10, and thus would have rendered that claim obvious.

- a. **Claim 10[pre]:** “A method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions”

227. See IX.A.7.a above.

- b. Claim 10[a]: “configuring the first processing module having a first memory as a master the provides requests”**

228. *See* IX.A.7.b, IX.C.1.b, above.

- c. Claim 10[b]: “configuring the second processing module having a second memory as a slave the provides responses to the requests”**

229. *See* IX.A.7.c, IX.C.1.b, above.

- d. Claim 10[c]: “connecting the master to a master interface unit of the interface units”**

230. *See* IX.A.7.d, above.

- e. Claim 10[d]: “connecting the master interface unit to the interconnect so that the master interface unit is between the master and the interconnect”**

231. *See* IX.A.7.e, above.

- f. Claim 10[e]: “connecting the slave to a slave interface unit of the interface units”**

232. *See* IX.A.7.f, above.

- g. Claim 10[f]: “connecting the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect”**

233. *See* IX.A.7.g, above.

- h. Claim 10[g]: “determining by a master determination unit of the master interface unit a first optimal amount of data to be buffered by a master wrapper of the master interface unit”**

234. *See* IX.A.7.h, above.

- i. **Claim 10[h]: “determining by a slave determination unit of the slave interface unit a second optimal amount of data to be buffered by a slave wrapper of the slave interface unit”**

235. *See* IX.A.7.i, above.

- j. **Claim 10[i]: “buffering by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered”**

236. *See* IX.A.7.j, above.

- k. **Claim 10[j]: “transferring the buffered data from the slave wrapper to the master wrapper when said first optimal amount of data has been buffered by the slave wrapper”**

237. *See* IX.A.7.k, above.

- l. **Claim 10[k]: “buffering by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper”**

238. *See* IX.A.7.l, above.

- m. **Claim 10[l]: “transferring the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper,”**

239. *See* IX.A.7.m, above.

- n. **Claim 10[m]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave,”**

240. *See* IX.A.7.n, above.

- o. **Claim 10[n]: “wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

241. *See* IX.A.7.o, IX.C.1.h, above.

8. Claim 1

242. In my opinion, Rădulescu in view of Rădulescu-WO and Bahadiroglu teaches all the limitations of claim 1 and thus would have rendered the claim obvious.

- a. **Claim 1[pre]: “An integrated circuit comprising:”**

243. *See* IX.C.1.a, above.

- b. Claim 1[a]: “a plurality of processing modules being connected with an interconnect and each of the plurality of processing modules having a memory, wherein a first processing module communicates to a second processing module using transactions, wherein the first processing module is a master for providing requests and the second processing module is a slave for providing responses to the requests;”**

244. *See* IX.C.7.a–IX.C.7.c, above.

- c. Claim 1[b]: “a first interface unit coupled to the slave and including a first wrapper for buffering data from the slave to be transferred over the interconnect to the master, the first interface unit further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first wrapper, and wherein the buffered data are transferred when said first optimal amount of data has been buffered by the first wrapper, and wherein the first interface unit including the first wrapper and the first determination unit is connected between the slave and the interconnect; and”**

245. *See* IX.C.7.f–IX.C.7.g, IX.C.7.i–IX.C.7.k, above.

- d. **Claim 1[c]: “a second interface unit coupled to the master and including a second wrapper for buffering data from the master to be transferred over the interconnect, the second interface unit further including a second determination unit, wherein second determination unit is configured to determine a second optimal amount of data to be buffered by the second wrapper, and wherein the buffered data are transferred when said second optimal amount of data has been buffered by the second wrapper, and wherein the second interface unit including the second wrapper and the second determination unit is connected between the master and the interconnect,”**

246. *See* IX.C.7.d–IX.C.7.e, IX.C.7.h, IX.C.7.l–IX.C.7.m, above.

- e. **Claim 1[d]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

247. *See* IX.C.7.n–IX.C.7.o , above.

9. **Claim 2: “The integrated circuit according to claim 1, wherein the first interface unit includes a first unbuffer-flag unit for providing a first unbuffer signal and the second interface unit includes a second unbuffer-flag unit for providing a second unbuffer signal, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to the first unbuffer signal and the second unbuffer signal, respectively, even if less than said first optimal amount and said second optimal amount of data is buffered in said first wrapper and said second wrapper.”**

248. *See* IX.C.8, IX.A.9, above.

10. **Claim 4: “The integrated circuit according to claim 2, wherein said first unbuffer signal is initiated by the slave and said second unbuffer signal is initiated by the master.”**

249. *See* IX.C.9, IX.A.10, above.

11. **Claim 6: “The integrated circuit according to claim 2, wherein the master is adapted to initiate said first unbuffer signal and said second unbuffer signal.”**

250. *See* IX.C.9, IX.A.10, above.

12. **Claim 8: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using messages of a message-passing communication scheme, wherein the messages contain a message header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said message header, respectively, even if less than said first amount and said second amount of data is buffered in said first wrapper and said second wrapper.”**

251. *See* IX.C.8, IX.A.12, above.

13. **Claim 9**

252. *See* IX.C.8, IX.A.13, above.

14. Claim 11

253. In my opinion, Rădulescu in view of Rădulescu-WO and Bahadiroglu teaches all the limitations of claim 11 and thus would have rendered the claim obvious.

a. Claim 11[pre]: “A data processing system, comprising:”

254. *See* IX.A.14.a, above.

b. Claim 11[a]: “a plurality of processing modules each having a memory and being connected with an interconnect, wherein a first processing module communicates to a second processing module using transactions, wherein the first processing module is a master for providing requests and the second processing module is a slave for providing responses to the requests; and”

255. *See* IX.C.8.b, above.

c. Claim 11[b]: “a first interface unit coupled to the slave and including a first wrapper for buffering data from the slave to be transferred over the interconnect, the first interface unit further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first wrapper, and wherein the buffered data are transferred when said first optimal amount of data has been buffered by the first wrapper, and wherein the first interface unit including the first wrapper and the first determination unit is connected between the slave and the interconnect; and”

256. *See* IX.C.8.c, above.

- d. **Claim 11[c]: “a second interface unit coupled to the master and including a second wrapper for buffering data from the master to be transferred over the interconnect, the second interface unit further including a second determination unit, wherein second first determination unit is configured to determine a second optimal amount of data to be buffered by the second wrapper, and wherein the buffered data are transferred when said second optimal amount of data has been buffered by the second wrapper, and wherein the second interface unit including the second wrapper and the second determination unit is connected between the master and the interconnect,”**

257. *See* IX.C.8.d, above.

- e. **Claim 11[d]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

258. *See* IX.C.8.e, above.

D. Ground 4: Claims 1, 2, 4, 6, 8–12, 13, and 18–21 are Rendered Obvious by Rădulescu in view of Petersen and Rădulescu-WO

259. In my opinion, claims 1, 2, 4, 6, 8–12, 13, and 18–21 are rendered obvious by the combination of Rădulescu, Petersen, and Rădulescu-WO.

260. A POSITA would have combined these references for all of the reasons that I explained in Grounds 2 and 3. A POSITA would also have had a reasonable expectation that the combination would succeed for the reasons that I explained previously.

1. Claim 12

261. In my opinion, Rădulescu in view of Petersen and Rădulescu-WO teaches all the limitations of claim 12 and thus would have rendered that claim obvious.

a. Claim 12[preamble]: “An integrated circuit comprising”

262. *See* IX.B.1.a, above.

b. Claim 12[a]: “a plurality of processing modules including a first processing module having a first processing memory and a second processing module having a second processing memory”

263. *See* IX.B.1.b, IX.C.1.b, above.

- c. **Claim 12[b]: “a plurality of interconnect modules including a first interconnect module and a second interconnect module, wherein the first processing module is connected to the first interconnect module and the second processing module is connected to the second interconnect module”**

264. *See* IX.B.1.c, above.

- d. **Claim 12[c]: “a network for connecting the first interconnect module to the second interconnect module in order to provide a communication connection between the first processing module and the second processing module, wherein the first interconnect module is connected between the first processing module and the network, and wherein the second interconnect module is connected between the second processing module and the network”**

265. *See* IX.C.1.d, above.

- e. **Claim 12[d]: “wherein the first interconnect module includes a first memory for buffering first data from the first processing module, the first interconnect module further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first memory, and the second interconnect module includes a second memory for buffering second data from the second processing module, the second interconnect module further including a second determination unit, wherein the second determination unit is configured to determine a second optimal amount of data to be buffered by the second memory”**

266. *See* IX.B.1.e, above.

- f. **Claim 12[e]: “the first interconnect module transferring the first data to the second processing module when the first data buffered in the first memory reaches the first optimal amount, and the second interconnect module transferring the second data to the first processing module when the second data buffered in the second memory reaches the second optimal amount”**

267. *See* IX.B.1.f, above.

- g. **Claim 12[f]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave”**

268. *See* IX.B.1.g, above.

- h. **Claim 12[g]: “wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery”**

269. *See* IX.B.1.h, IX.C.1.h, above.

- 2. **Claim 13: “The integrated circuit of claim 12, wherein the plurality of interconnect modules manages the communication connection freeing the plurality of processing modules to perform data processing.”**

270. *See* IX.D.1, IX.A.2, above.

3. **Claim 18:** “The integrated circuit of claim 12, wherein the first processing module provides a first flushing command to flush the first memory and a second flushing command to flush the second memory so that the first data and the second data are sent without waiting to reach the first amount and the second amount.”

271. *See* IX.D.1, IX.A.3, above.

4. **Claim 19:** “The integrated circuit of claim 18, wherein the first processing module provides the first flushing command and the second flushing command at least one of independently and simultaneously.”

272. *See* IX.D.3, IX.A.4, above.

5. **Claim 20:** “The integrated circuit of claim 18, wherein the second flushing command includes an indication to flush the second memory so that the first interconnect module forwards the second flushing command to the second interconnect module for flushing the second data from the second memory.”

273. *See* IX.D.3, IX.A.5, above.

6. **Claim 21:** “The integrated circuit of claim 12, wherein the first processing module includes flush information in a header of a message to cause all messages from a same connection that have been sent earlier to be flushed from at least one of the first memory and the second memory.”

274. *See* IX.D.1, IX.A.6, above.

7. **Claim 10**

275. In my opinion, Rădulescu in view of Petersen and Rădulescu-WO teaches all the limitations of claim 10 and thus would have rendered it obvious.

- a. **Claim 10[pre]: “A method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions”**

276. *See* IX.B.7.a, above.

- b. **Claim 10[a]: “configuring the first processing module having a first memory as a master the provides requests”**

277. *See* IX.B.7.b, IX.C.7.b, above.

- c. **Claim 10[b]: “configuring the second processing module having a second memory as a slave the provides responses to the requests”**

278. *See* IX.B.7.c, above, IX.C.7.b, above.

- d. **Claim 10[c]: “connecting the master to a master interface unit of the interface units”**

279. *See* IX.B.7.d, above.

- e. **Claim 10[d]: “connecting the master interface unit to the interconnect so that the master interface unit is between the master and the interconnect”**

280. *See* IX.B.7.e, above.

- f. **Claim 10[e]: “connecting the slave to a slave interface unit of the interface units”**

281. *See* IX.B.7.f, above.

- g. **Claim 10[f]: “connecting the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect”**

282. *See* IX.B.7.g, above.

- h. Claim 10[g]: “determining by a master determination unit of the master interface unit a first optimal amount of data to be buffered by a master wrapper of the master interface unit”**

283. *See* IX.B.7.h, above.

- i. Claim 10[h]: “determining by a slave determination unit of the slave interface unit a second optimal amount of data to be buffered by a slave wrapper of the slave interface unit”**

284. *See* IX.B.7.i, above.

- j. Claim 10[i]: “buffering by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered”**

285. *See* IX.B.7.j, above.

- k. Claim 10[j]: “transferring the buffered data from the slave wrapper to the master wrapper when said first optimal amount of data has been buffered by the slave wrapper”**

286. *See* IX.B.7.k, above.

- l. Claim 10[k]: “buffering by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper”**

287. *See* IX.B.7.l, above.

- m. **Claim 10[l]: “transferring the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper,”**

288. *See* IX.B.7.m, above.

- n. **Claim 10[m]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave,”**

289. *See* IX.B.7.n, above.

- o. **Claim 10[n]: “wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

290. *See* IX.B.7.o, IX.C.1.h, above

8. Claim 1

291. In my opinion, Rădulescu in view of Petersen and Rădulescu-WO teaches all the limitations of claim 1 and thus would have rendered the claim obvious.

a. Claim 1[pre]: “An integrated circuit comprising:”

292. *See* IX.D.1.a, above.

b. Claim 1[a]: “a plurality of processing modules being connected with an interconnect and each of the plurality of processing modules having a memory, wherein a first processing module communicates to a second processing module using transactions, wherein the first processing module is a master for providing requests and the second processing module is a slave for providing responses to the requests;”

293. *See* IX.D.7.a–IX.D.7.c, above.

c. Claim 1[b]: “a first interface unit coupled to the slave and including a first wrapper for buffering data from the slave to be transferred over the interconnect to the master, the first interface unit further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first wrapper, and wherein the buffered data are transferred when said first optimal amount of data has been buffered by the first wrapper, and wherein the first interface unit including the first wrapper and the first determination unit is connected between the slave and the interconnect; and”

294. *See* IX.D.7.f–IX.D.7.g, IX.D.7.i–IX.D.7.k, above.

- d. **Claim 1[c]: “a second interface unit coupled to the master and including a second wrapper for buffering data from the master to be transferred over the interconnect, the second interface unit further including a second determination unit, wherein second determination unit is configured to determine a second optimal amount of data to be buffered by the second wrapper, and wherein the buffered data are transferred when said second optimal amount of data has been buffered by the second wrapper, and wherein the second interface unit including the second wrapper and the second determination unit is connected between the master and the interconnect,”**

295. *See* IX.D.7.d–IX.D.7.e, IX.D.7.h, IX.D.7.l–IX.D.7.m, above.

- e. **Claim 1[d]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

296. *See* IX.D.7.n–IX.D.7.o, above.

9. **Claim 2: “The integrated circuit according to claim 1, wherein the first interface unit includes a first unbuffer-flag unit for providing a first unbuffer signal and the second interface unit includes a second unbuffer-flag unit for providing a second unbuffer signal, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to the first unbuffer signal and the second unbuffer signal, respectively, even if less than said first optimal amount and said second optimal amount of data is buffered in said first wrapper and said second wrapper.”**

297. *See* IX.D.8, IX.A.9, above.

10. **Claim 4: “The integrated circuit according to claim 2, wherein said first unbuffer signal is initiated by the slave and said second unbuffer signal is initiated by the master.”**

298. *See* IX.D.9, IX.A.10, above.

11. **Claim 6: “The integrated circuit according to claim 2, wherein the master is adapted to initiate said first unbuffer signal and said second unbuffer signal.”**

299. *See* IX.D.9, IX.A.10, above.

12. **Claim 8: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using messages of a message-passing communication scheme, wherein the messages contain a message header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said message header, respectively, even if less than said first amount and said second amount of data is buffered in said first wrapper and said second wrapper.”**

300. *See* IX.D.8, IX.A.12, above.

- 13. Claim 9: “The integrated circuit according to claim 1, wherein the master and the slave communicate with each other using packets of a packet based communication scheme, wherein the packets contain a packet header, and wherein said first wrapper and said second wrapper are adapted to transfer the buffered data in response to information in said packet header, respectively, even if less than said first amount and said second amount of data is buffered in said first and second wrapper.”**

301. See IX.D.8, IX.A.13, above.

14. Claim 11

302. In my opinion, Rădulescu in view of Rădulescu-WO and Petersen teaches all the limitations of claim 11 and thus would have rendered the claim obvious.

- a. Claim 11[pre]: “A data processing system, comprising:”**

303. See IX.A.14.a, above.

- b. Claim 11[a]: “a plurality of processing modules each having a memory and being connected with an interconnect, wherein a first processing module communicates to a second processing module using transactions, wherein the first processing module is a master for providing requests and the second processing module is a slave for providing responses to the requests; and”**

304. See IX.D.8.b, above.

- c. **Claim 11[b]: “a first interface unit coupled to the slave and including a first wrapper for buffering data from the slave to be transferred over the interconnect, the first interface unit further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first wrapper, and wherein the buffered data are transferred when said first optimal amount of data has been buffered by the first wrapper, and wherein the first interface unit including the first wrapper and the first determination unit is connected between the slave and the interconnect; and”**

305. *See* IX.D.8.c, above.

- d. **Claim 11[c]: “a second interface unit coupled to the master and including a second wrapper for buffering data from the master to be transferred over the interconnect, the second interface unit further including a second determination unit, wherein second first determination unit is configured to determine a second optimal amount of data to be buffered by the second wrapper, and wherein the buffered data are transferred when said second optimal amount of data has been buffered by the second wrapper, and wherein the second interface unit including the second wrapper and the second determination unit is connected between the master and the interconnect,”**

306. *See* IX.D.8.d, above.

- e. **Claim 11[d]: “wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.”**

307. *See* IX.D.8.e, above.

X. SECONDARY CONSIDERATIONS

308. I am unaware of any secondary considerations of non-obviousness that provide evidence that the Challenged Claims of the '800 patent are not obvious. Should Patent Owner assert any secondary considerations of non-obviousness, I reserve the right to respond at that time.

XI. CONCLUSION

309. I hereby declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct, and that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I understand that willful false statements are punishable by fine or imprisonment or both under Section 1001 of Title 18 of the U.S. Code.

Date: December 18, 2023

Respectfully submitted,

A handwritten signature in black ink, appearing to read "Todd Mowry", is written over a horizontal line.

Todd Mowry, Ph.D.