

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

QUALCOMM INCORPORATED, and
SAMSUNG ELECTRONICS CO., LTD.,
Petitioners

v.

NETWORK SYSTEM TECHNOLOGIES
Patent Owner

IPR2024-00355

U.S. Patent No. 8,072,893

**DECLARATION OF DR. MARWAN HASSOUN IN SUPPORT OF
PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 8,072,893**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	QUALIFICATIONS AND BACKGROUND.....	3
III.	MATERIALS REVIEWED AND SUMMARY OF OPINIONS	7
IV.	LEVEL OF ORDINARY SKILL IN THE ART	9
V.	LEGAL PRINCIPLES	11
VI.	TECHNICAL BACKGROUND	14
	A. Systems on Chip (“SoC”).....	14
	B. Network-on-Chips (“NoCs”).....	16
	C. Fixed-Length Packets	17
	D. Pipelined Wires.....	18
VII.	THE ’2893 PATENT, PROSECUTION HISTORY, AND CLAIMS	20
	A. ’2893 Patent Specification	20
	B. Prosecution History	23
	C. Claims.....	25
VIII.	CLAIM CONSTRUCTION	27
IX.	OVERVIEW OF THE PRIOR ART	30
	A. Bertozzi (Ex-1005).....	30
	B. Dielissen (Ex-1007).....	36
	C. Jantsch (Ex-1006).....	37
	D. Duato (Ex-1009).....	38
	E. Hassoun (Ex-1011).....	39
	F. Sullam (Ex-1013)	40
X.	THE PRIOR ART DISCLOSES OR SUGGESTS ALL OF THE FEATURES OF THE CHALLENGED CLAIMS	41
XI.	GROUND 1: BERTOZZI ANTICIPATE CLAIMS 1-3, 6-7, 10	42
XII.	GROUND 2: BERTOZZI alone or IN VIEW OF JANTSCH OR DUATO Renders claim 1-3, 6-7, and 10 obvious	105

XIII. GROUND 3: DIELISSSEN IN VIEW OF HASSOUN RENDERS CLAIMS 1-3, 6-7, 9-10 OBVIOUS	123
XIV. GROUND 4: GROUNDS 1-3 FURTHER IN VIEW OF SULLAM RENDERS CLAIM 8 OBVIOUS	191

I, Marwan Hassoun, declare as follows:

I. INTRODUCTION

1. I have been retained by Qualcomm Incorporated and Samsung Electronics Co., Ltd., (collectively “Petitioners”) as an independent expert consultant in this proceeding before the United States Patent and Trademark Office (“PTO”).

2. I am being compensated at a rate of \$650/hour for my time spent on this proceeding, which is my regular and customary rate for consulting work. I am being separately reimbursed for any out-of-pocket expenses. I am not receiving any other compensation.

3. My compensation is not contingent on the nature of my findings, the presentation of my findings in testimony or this declaration, or the outcome of this or any other proceeding. I also have no other interest in this proceeding.

4. I have been asked to consider whether certain references disclose and/or suggest the features recited in claims 1-3 and 6-10 of U.S. Patent No. 8,072,893 (“the ’2893 patent”) (Ex-1001), including considering what the prior art would disclose and/or teach and suggest to a person of ordinary skill in the art (“POSA”) at the time of the purported invention.

5. In this declaration, I may provide figures that I have annotated to aid in explaining my opinions. I note that these figures are not necessarily intended to

imply literal, precise physical or bodily combination or modification in exactly the way shown or annotated. I provide figures and accompanying annotations to help illustrate various concepts. They should not be taken as showing the only disclosure of a particular element, or the only way that a person of ordinary skill in the art would have modified or combined the prior art.

6. The opinions, their factual support, and the related discussion of background technology presented herein are true to the best of my knowledge and, if called to testify, I could and would testify competently as to them.

7. My analysis of the relevant materials is ongoing, and I will continue to review any new material as it is provided. This declaration represents only those opinions that I have formed to date. I reserve the right to revise, supplement, or amend my opinions below based on new information and on my continuing analysis of the materials already provided, and to respond to any testimony or other information provided by or on behalf of the owner of the patent at issue in this proceeding.

8. In this declaration I also added emphasis to various quotations and citations, I added highlighting to various excerpts. Unless I note otherwise, the emphasis added (e.g. bolding or italics) and highlighting is mine.

9. My opinions are set forth below.

II. QUALIFICATIONS AND BACKGROUND

10. I am an independent consultant for this matter. All the opinions I provide in this declaration are based on my own personal knowledge, experience, and professional judgment. In forming these opinions, I have relied on my years of education, research and experience in the field, as well as my review of the materials identified below and throughout this report. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein.

11. I set forth my education, experience, and qualifications in my resume, attached as Appendix B to this declaration. I also provide a summary below.

12. In 1983, I earned a bachelor's degree in electrical engineering from South Dakota State University. In 1984 and 1988, I earned a master's degree and a Ph.D., respectively, in Electrical Engineering from Purdue University with specialization in software and algorithms for computer-aided design of electronic circuits.

13. I was a Professor of Electrical and Computer Engineering at Iowa State University from 1988–2000, teaching and performing research in very-large-scale integration (VLSI) integrated circuits and was a Professor-of-the-Practice in the Electrical and Computer Engineering department at Tufts University teaching integrated circuits design from 2020-2021.

14. I am a named inventor on 56 issued patents and 3 pending patent applications related to highly integrated circuits with novel functionality in the areas of multi-gigabit serial transceivers, power conversion, memories, analog-to-digital converters (ADCs), digital-to-analog converters, RF devices, FPGAs, robotics, and digital signal processor filters. In addition, I have over 70 publications that include papers on high-speed interleaved analog-to-digital converters and digital-to-analog converters.

15. My work in industry and academia has included more than 30 years of experience in integrated circuits and systems-on-chip (SOC) design and products including hardware and software for digital and analog systems.

16. I am currently the President of Green Semiconductor, Inc. (d/b/a Intellectual Inspiration) and have over 30 years of experience in designing and developing a variety of technologies relating to hardware and software for systems and devices including computers and mobile devices.

17. I also hold a position at the Massachusetts Institute of Technology (MIT) as Director of Partnerships for the Sandbox Innovation Fund supervising and guiding startups in various areas of technology including hardware and software.

18. I have received or been nominated for several awards for my work, including the Innovation of the Year award at Xilinx, Inc.; the Distinguished

Engineer Award at South Dakota State University; the Faculty of the Year award at Iowa State University; and the Warren Bost Teaching Award at Iowa State University.

19. I am also a senior member of the Institute of Electrical and Electronics Engineers (IEEE).

20. I have held a variety of senior professional positions, including Senior Vice President of Engineering at KeyEye Communication, Inc.; Senior Director of Engineering for the Communications Technology Division at Xilinx, Inc.; President & Managing Partner in TraCHip, LLC; Co-Founder of MadMax Optics, Inc.; and Vice President of Wired Products at RocketChips, Inc. I have worked as a software engineer and a hardware engineer during my career.

21. I have been involved in building analog, digital, and mixed-signal integrated circuits products that involve communications, processor and co-processors, analog-to-digital converters, digital-to-analog converters, and serial and parallel interconnect (and their integration into large complex integrated SOCs), FPGAs, boards and multi-chip modules with multiple analog and digital subcomponents including on-chip and off-chip interconnect technologies.

22. During my tenure at RocketChips, Xilinx and KeyEye Communications, I was involved in building Integrated Circuits products that involve hardware and software for communications and processor and co-

processor design and integration on large complex integrated System-on-Chips (SOCs) with multiple analog and digital subcomponents. I have been involved in all aspects of integrated circuits product lifecycle, including planning, design, licensing, verification, integration, synthesis, physical design (including layout), testing, qualification, fabrication, software, debug, fixing, revisions, ECOs (Engineering Change Orders) and customer support. My involvement included performing these tasks myself, managing large engineering groups that cover the entire spectrum of the product cycle, and outsourcing to third parties. The chips developed went into various customer products such as computers, servers, network equipment and mobile devices.

23. I have authored several chapters in several books: The Circuits and Filters Handbook, first, second and third editions, and Analog CAD Tools: Symbolic Techniques And Applications in 1995, 2003, 2009 and 1998, respectively. The chapters I authored relate to the use of software algorithms to analyze various circuits and systems used in electronic devices. In addition to the examples provided above, additional relevant information related to my education and professional experiences is provided in my Curriculum Vitae and related materials.

24. The opinions, their factual support, and the related discussion of background technology presented herein are true to the best of my knowledge and, if called to testify, I could and would testify competently as to them.

25. My analysis of the relevant materials is ongoing, and I will continue to review any new material as it is provided. This declaration represents only those opinions that I have formed to date. I reserve the right to revise, supplement, or amend my opinions below based on new information and on my continuing analysis of the materials already provided, and to respond to any testimony or other information provided by or on behalf of the owner of the patent at issue in this proceeding.

III. MATERIALS REVIEWED AND SUMMARY OF OPINIONS

26. My opinions contained in this Declaration are based on my knowledge and experience gained from my education in the field of electrical engineering, my experience in the field of integrated circuit design in both academia and industry over the last 30+ years, and the specific references, documents, and other materials I discuss in this Declaration.

27. In forming my opinions, I also reviewed the following (along with any other materials I discuss in my Declaration):

- The '2893 Patent (Ex-1001)

- The prosecution history of the '2893 Patent (Ex-1002), and the references discussed therein
- Davide Bertozzi et al., "*Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip*," IEEE Circuits and System Magazine (Volume 4, Issue: 2, 2004) ("Bertozzi"; Ex-1005)
- Axel Jantsch and Hannu Tenhunen (Eds.), *Networks on Chip*, Kluwer Academic Publishers (2003) ("Jantsch"; Ex-1006)
- Jose Duato et al., *Interconnection Networks an Engineering Approach*, IEEE Computer Society Press (1997) ("Duato"; Ex-1009)
- John Dielissen et al., *Concepts and Implementation of the Philips Network-on-Chip*, IP-Based SOC Design, November 2003 ("Dielissen"; Ex-1007)
- Hassoun et al., *Optimal Buffered Routing Path Constructions for Single and Multiple Clock Domain Systems*, IEEE/ACM International Conference on Computer Aided Design (2002) ("Hassoun"; Ex-1011).

28. I have been asked to form the opinions discussed in this Declaration by placing myself in the position of a person of ordinary skill back in the early 2005 timeframe. I have done so. I note that it was easy to place myself in that frame of mind, as I was a person of ordinary skill in the art at that time. My

opinions would not change if the relevant timeframe was slightly earlier or slightly later.

29. I am also guided by my understanding of how a person having ordinary skill in the art would have understood the claims and the specification of the '2893 patent at the time of the alleged invention. My opinions reflect how a POSA would have understood the '2893 patent, the prior art to the patent, and the state of the art at the time of the alleged invention.

30. Based on my experience, it is my opinion that certain references I discuss below disclose and/or suggest all the features recited in claims 1-3 and 6-10 of the '2893 patent, as I explain in more detail in the following sections.

IV. LEVEL OF ORDINARY SKILL IN THE ART

31. I understand that, in the context of an invalidity analysis, a person having ordinary skill in the art is a hypothetical person who would be aware of the prior art at the time of the invention. I also understand that to determine the appropriate level of skill in the art, the following factors may be considered: (a) the types of problems encountered by those working in the field and prior art solutions thereto; (b) the sophistication of the technology in question and the rapidity with which inventions occur in the field; (c) the education level of active workers in the field; and (d) the education level of the inventor.

32. Applying these considerations, it is my opinion that the relevant POSA for the Accused Patents would typically have had at least either a Master's degree in Electrical Engineering, or a related field, as well as at least three years of work experience in the field of SoC design, or a Bachelor's degree in Electrical Engineering, or related field, as well as at least five years of work experience in the field of SoC design. This POSA would have had an understanding of SoC interconnects. They would also have a working understanding of the concepts of large-scale networks. I was at least a person of ordinary skill in this art in early 2005 (and well before then). My opinions set forth in this declaration would not change if the POSA is found to have a slightly higher or lower level of experience and education.

33. All my opinions in this declaration are from the perspective of one of ordinary skill in the art as I have defined it here during the relevant timeframe.

34. I am aware that in related litigation, Patent Owner has asserted the following description of a POSA for the patents at issue, including the '2893 patent:

[A] POSA 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.

My opinions would not change if that level of skill were applied to my analysis below.

V. LEGAL PRINCIPLES

35. I am not an attorney and I do not offer any legal opinions in this Declaration. However, in the course of my work in the field of SOC design, as a technical consultant to legal cases, and in the course of being involved as a co-inventor on many issued patents, I have had experience studying and analyzing patents and patent claims from the perspective of a person skilled in the art.

36. For the purposes of this declaration, I have been informed about certain aspects of the law that are relevant to forming my opinions. I summarize my understanding of the law below.

37. I have been informed that a patent claim is invalid as anticipated under 35 U.S.C. § 102 if each and every element of a claim, as properly construed, is found either expressly or inherently in a single prior art reference. Under what is known as inherency, I understand that if the prior art necessarily functions in accordance with, or includes the claimed limitations, it anticipates. I also understand that I may examine evidence that is outside of the prior art reference in

determining whether a feature, while not expressly discussed in the reference, is necessarily present in that reference.

38. I have been informed that a reference can anticipate a claim even if it does not expressly spell out all the limitations arranged or combined as in the claim, if a person of skill in the art, reading the reference, would “at once envisage” the claimed arrangement or combination.

39. I also understand that a patent claim is only valid if it is nonobvious. I understand that a patent claim can be considered to be obvious even if all of the requirements of a claim are not found in a single prior art reference, if the differences between the subject matter in the prior art and the subject matter in the claim would have been obvious to a person having ordinary skill in the art at the time of the invention.

40. I have been informed that the Supreme Court considered the legal standard for obviousness in its decision in *KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398 (2007). I understand that a patent claim is invalid as obvious under 35 U.S.C. § 103 if the subject matter would have been obvious to a POSA at the time the alleged invention was made. I understand that the analysis of obviousness requires consideration of three factors: (1) a determination of the scope and content of the prior art; (2) the level of ordinary skill in the art; and (3) the differences between the prior art and the asserted claims. I also understand that, when evidence is

presented, certain objective considerations are to be considered as part of the obviousness analysis.

41. I have been informed that, under KSR, claims directed to a combination of familiar elements according to known methods may be invalid as obvious when the combination does no more than yield predictable results.

42. I have been informed that a claim may be obvious in light of a single prior art reference or in light of the combination of multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, I understand there must be a reason to modify the single prior art reference, or combine two or more references, with a reasonable expectation of success, in order to achieve the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine. However, I understand that a specific teaching, suggestion, or motivation to combine elements in the prior art to achieve the claimed combination is not required for obviousness. Instead, I understand that a combination of known elements may be obvious in light of the interrelated teachings of the prior art, the effects of demands in the market, the background knowledge possessed by a POSA, and inferences and creative steps that a POSA would employ.

43. I also understand that a combination of known elements may be obvious when there is a design need or market pressure to solve a problem and

there are a finite number of identified, predictable solutions known to a person of ordinary skill, such that the combination would have been obvious to try.

44. I also understand that in considering obviousness, obviousness may not be determined using the benefit of hindsight, including hindsight derived from the patent being considered.

45. I also understand that what is known as evidence of secondary considerations or objective evidence of non-obviousness, such as commercial success of an embodiment of the invention, solving a long-felt need, or industry praise, is relevant to obviousness. I am not currently aware of any such evidence or purported evidence pertaining to the '2893 patent. If I do become aware of such evidence, I may offer opinions regarding that evidence.

VI. TECHNICAL BACKGROUND

46. In this section, I provide some background on technology as understood as of the priority date of the '2893 patent and earlier. I focus the discussion on some relevant features that were known by that time and which I view as relevant to the declaration regarding the '2893 patent.

A. Systems on Chip ("SoC")

47. Historically, electronic systems consisted largely of discrete components scattered across printed circuit boards. As semiconductor technologies advanced, it became possible to integrate more of these components

on a single chip. Miniaturization was desirable because it reduced the size of the system, reduced the power consumption of the system, and increased the speed at which the system could operate. Eventually, it became possible to integrate enough components onto a single chip that the term “System-on-Chip” was coined, referencing the fact that an entire system could be integrated onto a single chip.

Ex-1006 at 4 (“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).”). SoC technology integrated multiple functions, including processing cores, memory, peripherals, and specialized hardware (sometimes called “modules”), onto as few as a single semiconductor chip. *Id.* By the late 1990s, SoCs were well-known. *E.g., id.*

48. The modules in a SoC needed to communicate with each other.

Before the earliest priority date of the patents-in-suit, these communications were implemented in a variety of ways which were tailored to meet specific design requirements and constraints. These communication systems for SoCs were commonly referred to as interconnection networks, on-chip interconnects or simply interconnects. In the 2003 time frame, examples of known interconnection networks included point-to-point wires, buses, crossbars, switches, routers, communication fabrics, and network-on-chips.

B. Network-on-Chips (“NoCs”)

49. One type of interconnect developed for use in a system-on-chip was a network-on-chip, also known as a “NoC”. NoCs had already become popular prior to the filing of the ‘052 patent. *See* Ex-1016; Ex-1017; Ex-1018; Ex-1006; Ex-1036, 1:37–39 (“Networks on chip (NoC) have received considerable attention recently as a solution to the interconnect problem in highly-complex chips.”).

50. NoCs used the concepts from large-scale networks and applied them to embedded SoCs. Ex-1036, 1:62–67 (“Most of these topics have been already the subject of research in the field of local and wide area networks (computer networks) and as an interconnect for parallel machine interconnect networks. Both are very much related to on-chip networks, and many of the results in those fields are also applicable on chip.”).

51. Various papers in the early 2000s explained potential advantages offered by NoCs. *E.g.*, Ex-1006,13 (“The NoC paradigm is highly suited to provide SoC platforms scalable and adaptable over several technology generations because it opens the opportunity to define and standardize a communication service infrastructure and a protocol stack from the physical to the application layer.”); Ex-1009, 6 (“As designers strive to make more efficient use of scarce interconnection bandwidth, interconnection networks are emerging as a nearly universal solution to the system-level communication problems for modern digital

systems”); Ex-1036, 1:39-44 (“NoCs help resolve the electrical problems in new deep-submicron technologies, as they structure and manage global wires. At the same time they share wires, lowering their number and increasing their utilization. NoCs can also be energy efficient and reliable and are scalable compared to buses.”). Accordingly, NoCs can be a scalable and flexible approach to interconnecting the different functional blocks, such as processors, memory, and peripherals, within a single integrated circuit.

C. Fixed-Length Packets

52. In a packet switched network, one consideration was the whether the packets should be fixed-length or variable-length. Ex-1015, 1. Fixed-length packets (sometimes referred to as fixed-sized packets) each have the same size or length. *See* Ex-1015, 3 (“ATM packets are fixed-sized packets with data fields of 48 bytes. Five bytes are used for header, so the length of the whole packet is 53 bytes.”); *see also* Ex-1035, 4 (“fixed length: Pertaining to a record of field that has a constant length, regardless of the specific data contained in it. Filler characters may be used to maintain the fixed length. *Contrast:* variable length. *See also:* fixed format.”) and 3 (fixed format: A file organization in which all logical records in the file of fixed length. *Synonym:* F format. *Contrast:* variable format.”) (emphasis in original).

53. Variable-length packets (sometimes referred to as variable-sized packets) can have different lengths. *See* Ex-1015, 4 (“PTM packets are variable-sized packets, and in this paper, we consider packets with maximum data field of 4096 bytes....”); *see also* Ex-1035, 6 (variable length: Pertaining to a record or field that does not have a constant length, but whose length depends on the length of the specific data contained in it. *Contrast:* fixed length, *See also:* variable format.”) and 6 (“variable format: a filed organization in which logical records are of variable length. *Synonym:* V format. *Contrast:* fixed format.”). Various networks use fixed-length packets. One of the best-known examples is Asynchronous Transfer Mode (ATM). Ex-1015, 3.

D. Pipelined Wires

54. By early 2005, it was well-known that a signal can only travel so far in a given amount of time. If a wire is long enough, a signal might not be able to traverse the entire wire in a single clock cycle. One solution to this problem was to slow down the clock such that the clock cycle is long enough for the signal to travel over the slow wire.

55. A second solution to this problem, also well-known by early 2005, was to “pipeline” the long wire with repeaters. *See* Ex-1005, 5, 7-9; Ex-1010, 1-2; Ex-1011, 1-3. The repeaters functioned as way-stations, breaking up the long wire into segments that the signal can travel over in a single clock cycle. Ex-1005, 7-8.

The use of pipelined wires introduces latency. *Id.* Instead of taking a single clock cycle to traverse the wire, it takes one additional clock cycle per repeater. *See id.*; *see also* Ex-1011, 3. These additional clock cycles are referred to as delay. *See* Ex-1011, 3. *See, e.g.,* Ex-1011, 2-3 (“We now explore the problem of finding a buffered routing path from s to t when ***multiple clock cycles are required. Routing over large distances*** in increasingly aggressive technologies will ***require several clock cycles*** to cross the die. Hence, ***one must periodically clock the signal by inserting synchronization elements (such as registers) along the signal path.*** In this case, one cannot simply treat a register like a buffer and add the register delay to the existing path delay in the Fast Path algorithm. The ***realizable delay between consecutive registers on a path will always be determined by the clock period, regardless of the actual signal propagation time. Register-to-register sub-paths with delays larger than the permissible clock cycle are not permitted.***”)

56. Dall’Osso teaches an example of NoC Architecture that includes links that “can be pipelined with a flexible number of stages to decouple data introduction speed from worst-case link delay.” Ex-1010, 1. Figure 2, reproduced below, shows such a pipelined link:

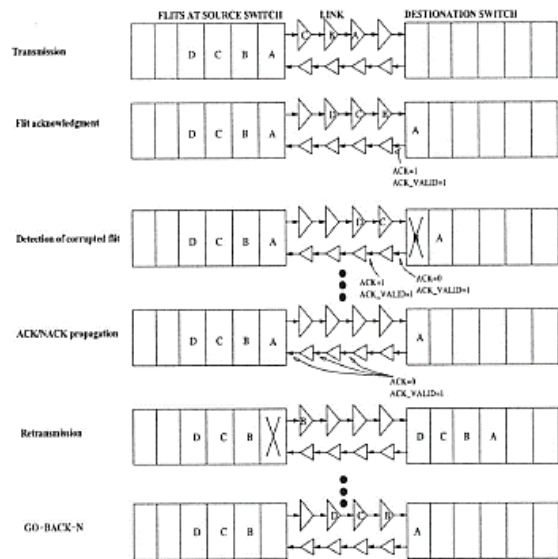


Figure 2: Pipelined switch-to-switch link model and its effect on retransmission of corrupted flits.

Ex-1010, Fig. 2.

VII. THE '2893 PATENT, PROSECUTION HISTORY, AND CLAIMS

A. '2893 Patent Specification

57. The '2893 patent is entitled “Integrated Circuit with Data Communication Network and IC Design Method”. The patent describes an integrated circuit including “functional blocks and a data communication network having network stations interconnected via communication channels for communicating data packages between the functional blocks.” Ex-1001, Abstract.

58. The '2893 patent explains that NoCs are used to “facilitate flexible communication on the one hand and to limit the amount of wires required on the other hand.” Ex-1001, 1:19-22. Such NoCs “typically comprise a plurality of network stations, each associated with a functional block, interconnected via

communication channels, e.g. wires.” *Id.*, 1:22-25. Network stations include data routers and network interfaces. *Id.*, 1:25-28. “The data can be communicated over the network in the form of packages that include a mixture of data and routing instructions for the network stations.” *Id.*, 1:28-30.

59. The ’2893 patent discloses the that a “problem associated with communication channels, e.g. wires, on ICs is that the clock speed at which the IC can operate may be determined by the slowest communication channel, i.e. the channel exhibiting the largest data transfer delay.” Ex-1001, 1:34-38. The ’2893 patent discusses a known solution to this problem is to “introduce a data storage element such as a latch into the slowest data communication channel.” *Id.*, 1:41-46. In that way “the clock speed of the data communication part of the IC can be increased at the expense of an additional clock cycle for the communication along the slowest communication channel.” *Id.*

60. However, the ’2893 patent teaches that this solution causes issues in network stations, such as routers, that are designed to “receive[] the routing data incorporated in the data packet during a predefined clock cycle.” Ex-1001, 1:47-57. “Delaying the communication to that router over a slow communication channel by a clock cycle to speed up the rest of the network would cause the routing data to arrive outside the predefined clock cycle, causing erroneous behaviour of the network.” *Id.*, 1:57-61.

61. In order to solve this problem, the '2893 patent teaches to insert “ $M*N$ data storage elements...for introducing a delay of $M*N$ cycles on the” communication channel, where “each data package compris[es] N data elements” and M is “a positive integer,” in order to maintain “the correct periodicity between the network stations.” *See, e.g.*, Ex-1001, 1:65-2:17, 3:50-51, 7:3-40; Figs. 1-4.

62. Figure 1, which I reproduce below, “network 100...is arranged to communicate data packages with size $N=3$.” *Id.*, 5:63-64. It shows a communication channel 150 between routers 120 and 140. Ex-1001, 5:5-10. Communication channel 150 comprises a first subchannel 152 and a second subchannel 154 between the respective inputs and outputs of routers 120 and 140. *Id.* “The network 100...is arranged to communicate data packages with size $N=3$.” *Id.*, 5:63-64. Figure 1 shows that 3 data storage elements 160 are inserted into both the first subchannel 152 and the second subchannel 154 (highlighted in yellow), such that $M*N$ data storage elements have been inserted into communication channel 150, where $M=2$ and $N=3$. *Id.*, 5:65-6:2. These data storage elements introduce a delay of 3 cycles on each subchannel. The data storage elements introduce a total delay of $M*N$ (where $M=2$ and $N=3$) cycles onto the communications channel. *Id.*, 5:49-62.

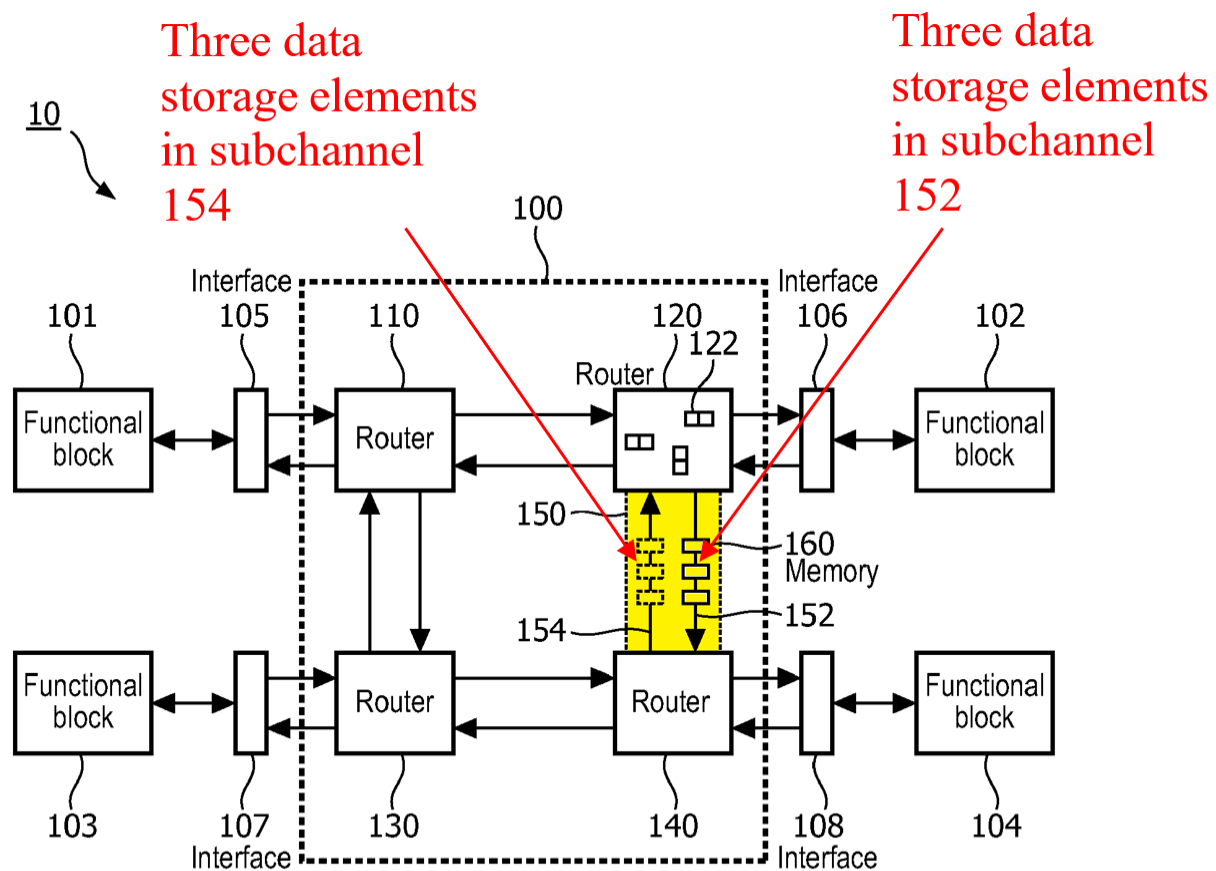


FIG. 1

Ex-1001, Fig. 1 (annotations added) and 5:5-10, 5:65-6:2. These data storage elements introduce a delay of 3 cycles on each subchannel. The data storage elements introduce a total delay of $M \cdot N$ (where $M=2$ and $N=3$) cycles onto the communications channel with half of that delay (3 cycles) on each of the two subchannels 152 and 154. *Id.*, 5:49-6:6.

B. Prosecution History

63. I have reviewed the file history for the '2893 patent. Below I provide a brief summary of my review.

64. During prosecution, the Examiner first rejected the several claims under 35 U.S.C. §103(a) as unpatentable over U.S. Patent No. 7,362,763 (“Wybenga”) in view of U.S. Patent No 7,197,009 (“Dispensa”) while objecting to some dependent claims that had patentable subject matter. Ex-1002, 125-134. In response, the patentee made amendments and argued against the rejection. Ex-1002, 144-165.

65. The Examiner then rejected claims under U.S. Patent Application No. 2002/0087716 (“Mustafa”) alone or in view of in various combinations with U.S. Patent Application No. 2004/0258099 (“Scott”) and U.S. Patent Application No. 2004/0193709 (“Selvaggi”) while objecting to some dependent claim that had patentable subject matter. Ex-1002, 169-185. In response, the patentee made amendments, including the “in response to the identifying act” language in claim 10 and the “when the data communication network identifies the first communication channel as having a data transfer delay exceeding a predefined delay threshold” language in claim 1, and made arguments against the rejection. Ex-1002, 194-206. The amended claims were then allowed by the Examiner. Ex-1002, 211-220.

C. Claims

66. The '2893 patent as issued contains five independent claims—claim 1, 4, 7, 10, and 11. I provide opinions regarding the invalidity of claims 1-3 and 6-10 below.

67. Independent claim 10 is a method of designing an integrated circuit claim that recites the following:

10. A method of designing an integrated circuit comprising a plurality of functional blocks, and a data communication network comprising a plurality of network stations being interconnected via a plurality of communication channels for communicating data packages between the functional blocks, each data package comprising N data elements including a data element comprising routing information for the network stations, N being an integer of at least two, the plurality of network stations comprising a plurality of data routers and a plurality of network interfaces, each of the data routers being coupled to a functional block via a network interface; the method comprising the acts of:

identifying a first communication channel between a first network station and a second network station that has a data transfer delay exceeding a predefined delay threshold; and

in response to the identifying act, inserting $M \times N$ data storage elements into the data communication network, M being a positive integer, for introducing a delay of $M \times N$ cycles on the first communication channel.

Claim 1 is an integrated circuit apparatus claim that recites the following:

1. An integrated circuit comprising:

a plurality of functional blocks; and

a data communication network comprising a plurality of network stations being interconnected via a plurality of communication channels for communicating data packages between the functional blocks, each data package comprising N data elements including a data element

comprising routing information for the network stations, N being an integer of at least two, the plurality of network stations comprising a plurality of data routers and a plurality of network interfaces, each of the data routers being coupled to a functional block via a network interface, the data communication network comprising a first network station and a second network station interconnected through a first communication channel, the data communication network further comprising $M*N$ data storage elements, M being a positive integer, the data communication introducing a delay of $M*N$ cycles on the first communication channel when the data communication network identifies the first communication channel as having a data transfer delay exceeding a predefined delay threshold.

Claim 7 is an integrated circuit apparatus claim that recites the following:

7. An integrated circuit comprising:

a plurality of functional blocks; and

a data communication network comprising a plurality of network stations being interconnected via a plurality of communication channels for communicating data packages between the functional blocks, each data package comprising N data elements including a data element comprising routing information for the network stations, N being an integer of at least two, the plurality of network stations comprising a plurality of data routers and a plurality of network interfaces, each of the data routers being coupled to a functional block via a network interface, the data communication network comprising a first network station and a second network station interconnected through a first communication channel, the data communication network further comprising $M*N$ data storage elements, M being a positive integer, for introducing a delay of $M*N$ cycles on the first communication channel,

wherein each communication channel of the plurality of communication channels comprises:

a first subchannel between an input port of a first associated network station and an output port of a second associated network station; and

a second subchannel between an input port of the second associated network station and an output port of the first associated network station, and

wherein the first subchannel and the second subchannel of the first communication channel each comprise a delay introduced by $\frac{1}{2} * M * N$ data storage elements.

68. I have provided an appendix listing the claim elements, for reference, in Appendix A.

VIII. CLAIM CONSTRUCTION

69. I understand that Petitioner Qualcomm and Patent Owner have submitted claim constructions in *Network System Technologies, LLC v. Qualcomm Inc. et al.*, No. 1:22-cv-01331 (W.D. Tex.). I have reproduced these constructions below:

Term for Construction	Petitioner Qualcomm's Proposed Construction	Patent Owner's Proposed Construction
"each data package comprising N data elements"	"every data package communicated by the network stations has a fixed size of N data elements"	NST incorporates its definition for the term "data package," "data elements" are groups of data, and the remaining language should be afforded their plain and ordinary meanings.
"data package"	See Defendants' construction of "each data package comprising N data elements" above.	"Fixed-size arrangement of data"
"data storage elements"	"each being a structure that stores one data element"	"Array of one or more elements that store data."

Ex-1025; Ex-1026; Ex-1027. Though I agree with Petitioner Qualcomm's constructions and disagree with Patent Owner's constructions, I have applied both Petitioner Qualcomm's Constructions and Patent Owner's constructions where relevant because it is my opinion that the references I discuss teach the element under either one.

70. I have reviewed Patent Owner's claim construction brief (Ex-1026) from *Network System Technologies, LLC v. Qualcomm Inc. et al.*, No. 1:22-cv-01331 (W.D. Tex.). In that brief, Patent Owner provided additional discussion of its constructions of "data package" and "each data package comprising N data elements." Ex-1026, 30-34. The Patent Owner explained that "fixed size" in its construction merely refers to "the fact that the number of data elements in a given data package must remain fixed (*i.e.*, constant) throughout the course of each communication. While the number of data elements can vary between one data package and the next, the number for any given data package cannot fluctuate (*i.e.*, must remain 'fixed')." Ex-1026, 31. Patent Owner explained that under its construction, the number of data elements in each data package traveling through the system "can vary between one data package and the next." *Id.* The Patent Owner's expert also stated that under the Patent Owner's construction "the term 'each data package comprising N data elements' means that each data package

comprises N, or more than N, data elements, i.e., at least N data elements.” Ex-1027, ¶164.

71. I understand Patent Owner’s Construction of “data storage elements” is broader than Petitioner Qualcomm’s Construction. Any “data storage element” satisfying Petitioner Qualcomm’s Construction would also satisfy Patent Owner’s Construction. An “Array of one or more elements that store data” includes multiples of “a structure that stores one data element.” Patent Owner defined array as “a group of elements forming a complete unit.” Ex-1026, 34.

72. I have been told that Patent Owner has interpreted element [1.f] to have the same scope as claim elements [10.e] and [10.f] in *Network System Technologies, LLC v. Samsung Electronics Co, Ltd.*, No. 2:22-cv-481 (E.D. Tex.). I disagree with this conclusion. However, I have been instructed to apply the same scope to element [1.f] as elements [10.e] and [10.f], and I have done so.

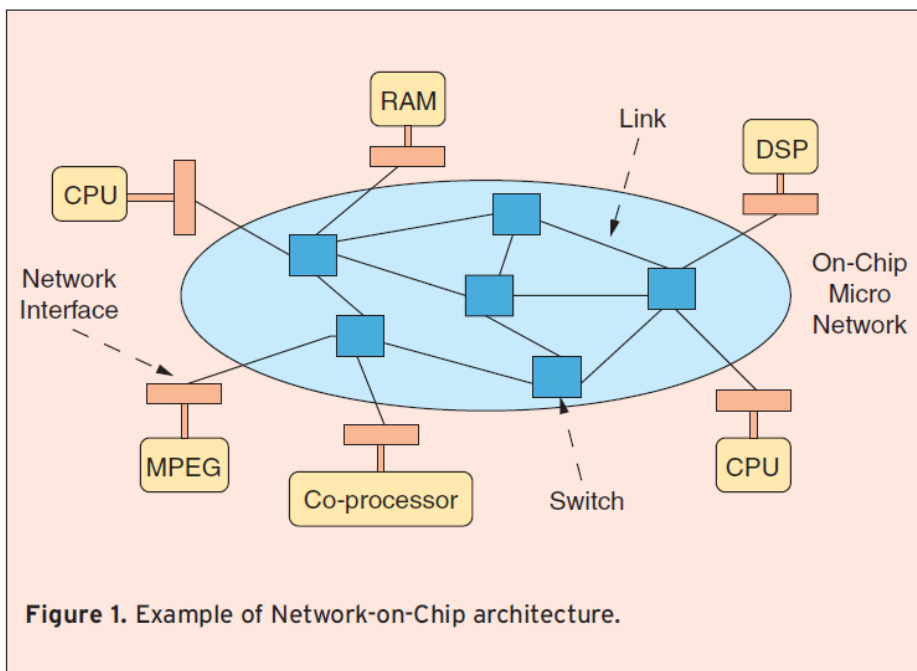
73. For all other terms, I have given the claim terms their plain and ordinary meaning, as would have been understood by a person having ordinary skill in the art, at the time of the alleged invention, which I am informed is approximately early 2005 or slightly earlier, having considered the language of the claims, the specification, and the prosecution history of record.

IX. OVERVIEW OF THE PRIOR ART

A. Bertozzi (Ex-1005)

74. Bertozzi discloses a “Network-on-Chip” (NoC) design architecture call “Xpipes.” Ex-1005, Title, Abstract, 1-2. The Xpipes NoC allows for the communication of data packets between cores (CPUs, RAM, etc.). Ex-1005, 2 and Fig. 1.

75. Bertozzi discloses a general “Network-on-Chip” architecture for use in its Xpipes designs. This is explained generally at Ex-1005, 2-3. For example, Bertozzi states that “[t]he basic idea . . . envisions on-chip router (or switch)-based networks on which packetized communication takes place.” *Id.* at 2. As another example, Bertozzi explains that “[w]hile describing the basic building blocks of [the Xpipes] solution, the main NoC design principles will be briefly explained.” *Id.* at 33. At Figure 1 (which I reproduced below), Bertozzi depicts these components with a NoC within an integrated circuit. *See also, infra* ¶¶94, 95.



Ex-1005, Fig. 1. Bertozzi explains the various NoC components depicted in this figure, including the network interfaces, switches and links. Ex-1005, 2, 5, 7-11, Figs. 4, 5, 7. To a POSA, Bertozzi’s teaching indicates that this basic NoC architecture is used for the Xpipes implementation discussed in the Bertozzi paper.

76. Bertozzi also provides additional general design features of “data packetization” for NoC data packets that are for use in Bertozzi’s Xpipes. *E.g.*, Ex-1005, 3 (“While describing the basic building blocks of this solution, the main NoC design principles will be briefly explained, detailing the specific contribution of Xpipes . . .”), 4-5 (“most relevant tasks of a *network interface (NI)* are . . . data packetization”). Bertozzi explains that “data packetization is a critical task for the network interface” and that messages “are usually partitioned into fixed-length packets.” Ex-1005, 5. Bertozzi further describes the data packets as follows:

Data packetization is a critical task for the network interface, and has an impact on the communication latency, in addition to the latency of the communication channel. Messages that have to be transmitted across the network are usually partitioned into fixed-length packets. Packets in turn are often broken into message flow control units called *flits*. In the presence of channel width constraints, multiple physical channel cycles can be used to transfer a single flit. A *phit* is the unit of information that can be transferred across a physical channel in a single step. Flits represent logical units of information, as opposed to phits that correspond to physical quantities. In many implementations, a flit is set to be equal to a phit.

The packet preparation process consists of building the packet header, payload and packet tail. The header contains the necessary routing and network control information (e.g., the source and destination address). When source routing is used, the destination address is ignored and replaced with a route field that specifies the route to destination. This overhead in terms of packet header is counterbalanced by a simpler routing logic at the network switches: they simply have to look at the routing fields in the packets and route them over the specified switch output port. The packet tail indicates the end of a packet and usually contains parity bits for error-detecting or error-correcting codes. Packetizing data deals effectively with communication errors. In fact, packet boundaries contain the effect of errors and allow error recovery on a packet-by-packet basis.

The task of the switch is to carry packets injected into the network to their final destination, following a statically defined or dynamically determined routing path. The switch transfers packets from one of its input ports to one or more of its output ports.

Ex-1005, 5.

77. Figure 4 of Bertozzi represents a specific embodiment of a data packet having four flits A, B, C, and D transmitted between two successive switches. Ex-1005, 8, Fig. 4. A POSA would understand Figure 4 to be showing a data packet with four data elements. The flits, which are equal to a phit (a data element) in

Bertozzi's implementation, are lettered A through D and are stored within a switch. The switch shows no additional data elements following, and shows a blank space after the last data element. Ex-1005, Fig. 4; 6 (Xpipes is a "packet-switched NoC[]"). Due to all of this, a POSA looking at Figure 4 would understand the flits lettered A, B, C, D to be a data packet of four data elements.

78. I note that Bertozzi also shows an Xpipes embodiment for a data packet with ten flits, at Figure 3. A POSA would understand that this is a general illustration of the partitioning procedure of a packet ("***Data packetization results in the packet partitioning procedure illustrated in Fig. 3,***" Ex-1005, 6) and is a different embodiment than the specific embodiment shown in Figure 4 that represents transmission between two consecutive switches (*see* previous paragraph).

79. Bertozzi also explains that "[a] solution to overcome the interconnect-delay problem consists of pipelining interconnects." Ex-1005, 7. Applying that solution, the Xpipes design includes links that are "partitioned into segments (or relay stations, which have a function similar to the one of latches in a pipelined data path)" to break up long links into shorter lengths that "satisf[y] pre-defined timing requirements (e.g., the desired clock speed of the design)." Ex-1005, 7. As Bertozzi explains:

Network Link

A solution to overcome the interconnect-delay problem consists of pipelining interconnects [34, 35]. Wires can be partitioned into segments (or relay stations, which have a function similar to the one of latches in a pipelined data path) whose length satisfies pre-defined timing requirements (e.g., the desired clock speed of the design). In this way, link delay is changed into latency, but the data introduction rate is not bound by the link delay any more. Now the latency of a channel connecting two modules may end up being more than one clock

cycle. This requires the system to be composed of modules whose behavior does not depend on the latency of input communication channels [34].

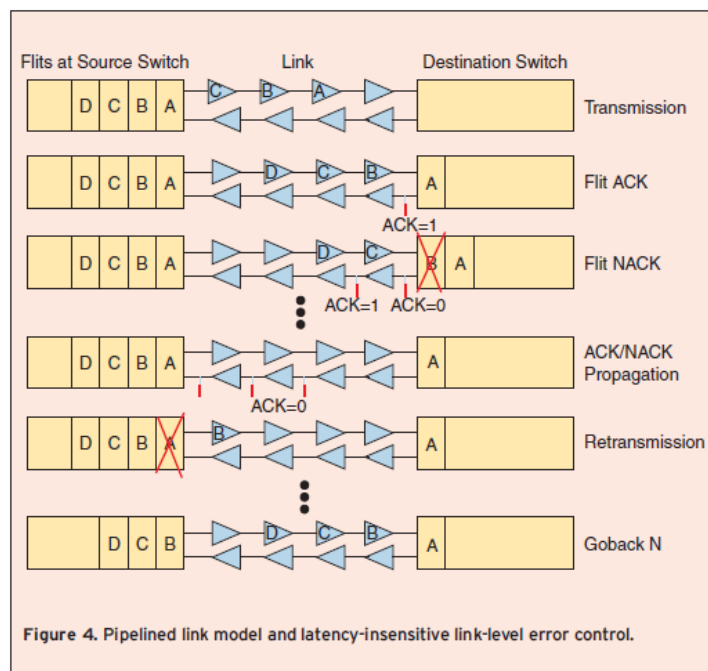
Xpipes interconnect makes use of pipelined links and of latency-insensitive operation of its building blocks. Switch-to-switch links are subdivided into basic segments whose length can be tailored to the desired clock frequency that needs to be achieved in the design. In this way, the system operating frequency is not bound by the delay of long links. According to the link length, a certain number of clock cycles is needed by a flit to cross the interconnect. If network switches are designed in such a way that their functional correctness depends on the flit arriving order and not on their timing, the input links of the switches can be different and of

any length. These design choices are at the basis of latency insensitive operation of the NoC and allow the construction of an arbitrary network topology and hence support for heterogeneous architectures.

Fig. 4 illustrates the link model, which is equivalent to a pipelined shift register. Pipelining has been used both for data and control lines. The figure also illustrates how pipelined links affect the operation of latency-insensitive link-level error control: the retransmission of a corrupted flit between two successive switches is represented. Multiple outstanding flits propagate across the link during the same clock cycle. When flits are correctly received at the destination switch, an ACK is propagated back to the source, and after N clock cycles (where N is the link length expressed in terms of number of repeater stages) the flit will be discarded from the buffer of the source switch. On the contrary, a corrupted flit is NACKed and will be retransmitted in due time.

Ex-1005, 5-6.

80. Figure 4, a specific embodiment of Bertozzi's Xpipes architecture, is below, showing pipelined links designed to have four repeaters in each subchannel:



Ex-1005, Fig. 4 (showing four repeaters in each subchannel).

81. A POSA would understand that Bertozzi's repeaters or relay stations are data storage elements. Ex-1005, 7 ("Wires can be partitioned into segments (or *relay stations, which have a function similar to the one of latches in a pipelined data path*) whose length satisfies pre-defined timing requirements (e.g., *the desired clock speed of the design*)."). A POSA would also understand that each repeater introduces a delay of one clock cycle on the channel. Ex-1005, 7-8 ("*Xpipes* interconnect *makes use of pipelined links*.... Switch-to-switch links are subdivided into *basic segments* whose length can be tailored *to the desired clock frequency* that needs to be achieved in the design. In this way, the system

operating frequency is not bound by the delay of long links. According to the link length, *a certain number of clock cycles is needed* by a flit to cross the interconnect...Fig. 4 illustrates the link model, which is *equivalent to a pipelined shift register*. *Pipelining has been used* both for data and control lines.”).

B. Dielissen (Ex-1007)

82. Dielissen teaches the “Aethereal” NoC. Ex-1007, 1. The Aethereal NoC “consists of routers and network interfaces.” Ex-1007, 1. IP modules are “seamlessly connect[ed]...to the NoC” via network interfaces. *Id.* “The routers transport packets of data from one NI to another. *Id.* Figure 1, reproduced below, shows the routers and network interfaces:

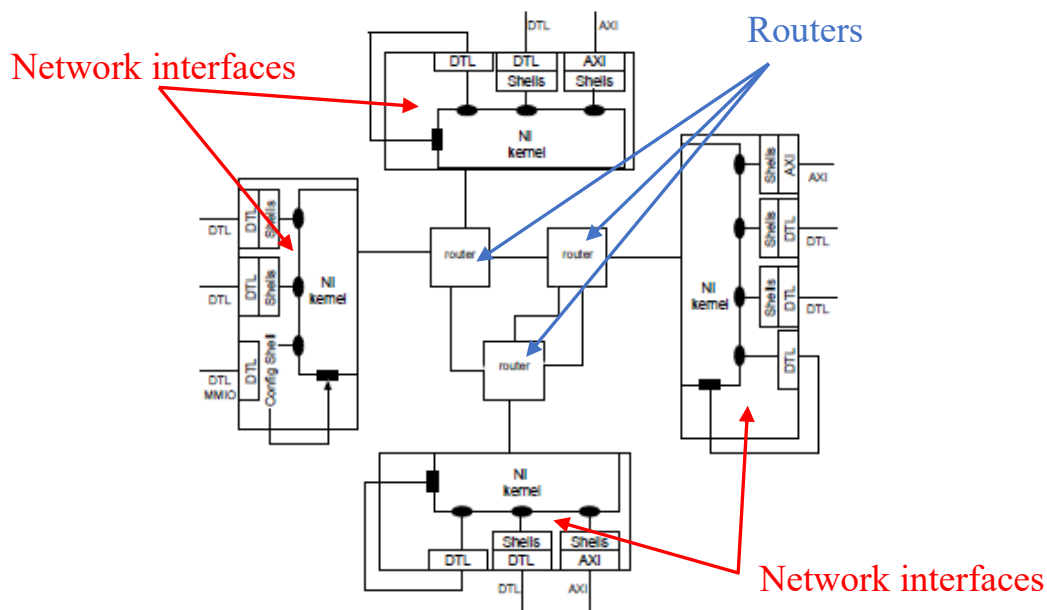


Figure 1. network example

Ex-1007, Fig. 1 (annotated).

83. Dielissen teaches that its packets are made up of “one or more flits.”

Ex-1007, 2. Each flit in turn contains three words. *Id.* Figure 4 illustrates such a packet:

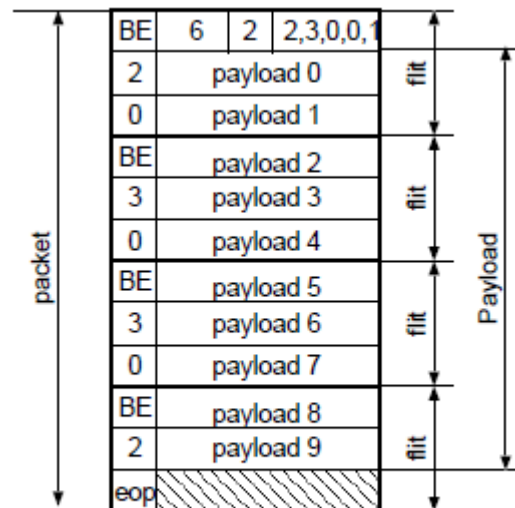


Figure 4. BE-packet example

Ex-1007, Fig. 4.

C. Jantsch (Ex-1006)

84. Jantsch is a textbook titled “Networks on Chip.” Ex-1006. It contains a collection of chapters by various authors. Of particular relevance is Chapter 5, entitled “On Packet Switched Networks for On-Chip Communication.” Ex-1006, 47. This chapter discusses the known approach of borrowing and adapting “the concepts of packet switched communication from computer networks to design on-chip networks.” Ex-1006, 47.

85. One such concept that Jantsch borrows from packet switched computer networks is the idea of “fixed packet size.” Ex-1006, 56. It discusses two

options of designing a NoC system: fixed packet size and variable packet size. *Id.* It recommends the “choosing a fixed packet size” because it will be “efficient as compared to variable packet size.” Ex-1006, 56.

D. Duato (Ex-1009)

86. Duato is another textbook titled “Interconnection Networks: an Engineering Approach. Ex-1009. Duato teaches various types of interconnection networks for use in many different applications, including “internal buses in very large-scale integration (VLSI) circuits.” Ex-1009, 13. This includes various forms of message switching, including designing networks made up of routers. *E.g.*, Ex-1009, 31-32.

87. Duato teaches designing such networks for “fixed-length packets.” Ex-1009, 32. It discusses numerous advantages in choosing to design a system to use fixed-length packets. For example, two such benefits are: reduced congestion resulting from long packets and improved deadlock recovery:

When traffic consists of messages of *very different lengths, a few long messages may reserve all of the virtual channels of a given physical channel*. If a short message requests that channel, it will have to wait for long. *A simple solution consists of splitting long messages into fixed length packets*. This approach reduces waiting time for short messages but introduces some overhead because each packet carries routing information and has to be routed.

Ex-1009, 98-99.

However, when long packets become deadlocked recovering from deadlock may take a while. As a consequence, other deadlocked packets have to wait a considerable amount of time before the central buffer becomes available, increasing latency and reducing throughput. In these situations, latency becomes much less predictable. Additionally, the optimal value of the timeout threshold for deadlock detection heavily depends on packet length unless some type of physical channel monitoring of neighboring nodes is implemented. *Therefore, short messages or messages split into fixed-length shorter packets may be preferred.*

Ex-1009, 67.

E. Hassoun (Ex-1011)

88. Hassoun discloses the use of buffered routing paths on wires when “multiple clock cycles are required” for a long signal path. Ex-1011, 2. Such signal paths “must periodically clock the signal by inserting synchronization elements (such as registers) along the signal path.” Ex-1011, 2-3. “Register-to-register sub-paths” cannot have delays that exceed “the permissible clock cycle,” a predefined delay threshold. Ex-1011, 3; *see also supra*, ¶55.

89. Such buffered routing paths contain a number of registers. Ex-1011, 3. Each such register results in a clock cycle of delay on the path. *Id.* (“The realizable delay between consecutive registers on a path will always be determined by the clock period”). For example, Figure 2 reproduced below, shows an example of

such a path between s and t with three registers, thereby introducing a delay of three cycles:

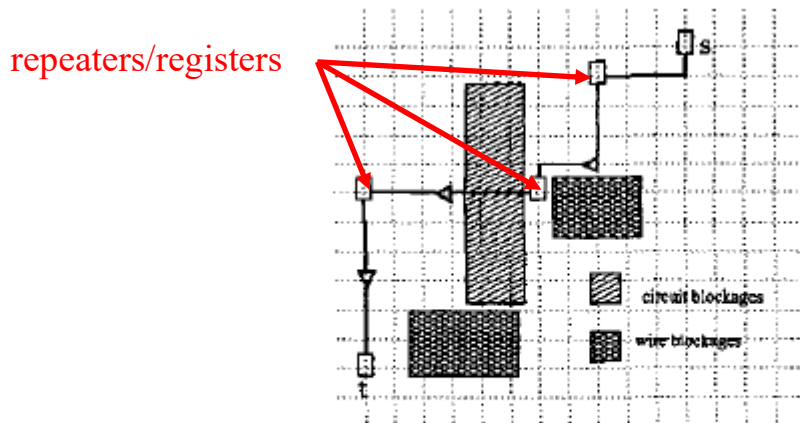


Figure 2. An example of routing within a single clock domain

Ex-1011, Fig. 2 (annotated), 3. “It thus takes four clock cycles to traverse from s to t....” Ex-1011, 3.

F. Sullam (Ex-1013)

90. Sullam discloses on-chip clocking circuitry. Ex-1013, 6:15-16 (“The microcontroller 100 of FIG. 1 also includes clocking circuitry 112”); Fig. 1. “[C]locking architecture 112 is able to provide[] a wide variety of clocking signal frequencies to different circuitry that it is couple to.” Ex-1013, 6:18-21. The clocking architecture includes an on-chip oscillator. Ex-1013, 7:35-37 (“The clocking architecture 112 also includes the 24 MHz oscillator 212 which may be

implemented with a precision integrated circuit (IC) oscillator.”). A POSA would understand the clocking circuitry taught by Sullam to be a clock signal generator.

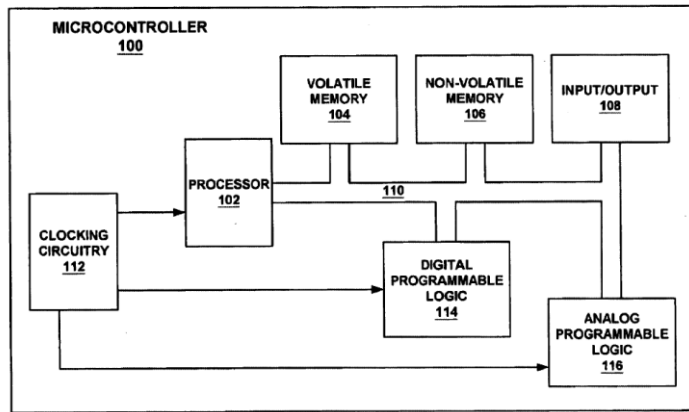


FIGURE 1

Ex-1013, Fig. 1.

X. THE PRIOR ART DISCLOSES OR SUGGESTS ALL OF THE FEATURES OF THE CHALLENGED CLAIMS

91. Below, I provide my opinions regarding several different grounds that challenge claims of the '2893 patent. These grounds are:

Ground	Claims	Basis for Unpatentability
Ground 1	1-3, 6-7, 10	Anticipation by Bertozzi
Ground 2	1-3, 6-7, 10	Obvious over Bertozzi alone, or in further view of Jantsch or Duato
Ground 3	1-3, 6-7, 9-10	Obvious over Dielissen in view of Hassoun
Ground 4	8	Obvious in view of Grounds 1-3, further in view of Sullam

92. Based on my experience, knowledge, and expertise, it is my opinion that certain references disclose and/or suggest all the features recited in each of claims 1-3 and 6-10 of the '2893 patent. I explain this in detail below.

XI. GROUND 1: BERTOZZI ANTICIPATE CLAIMS 1-3, 6-7, 10

93. Bertozzi discloses all the limitations of claim 10.

94. [10.pre]

Claim language	Evidence and disclosure
[10.pre] A method of designing an integrated circuit comprising	Bertozzi discloses this element. Bertozzi discloses a method for designing an integrated circuit. Bertozzi teaches the use of the Xpipes architecture and platform to design a NoC interconnect in a System-on-Chip (“SoCs”). <i>E.g.</i>, • Ex-1005, Title (“Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip.”); • Ex-1005, 1 (“In this paper, an advanced NoC architecture, called Xpipes, targeting high performance and reliable communication for on-chip multi-processors is introduced.”); • Ex-1005, 2 (“<i>The integration of an entire system onto the same silicon die</i> (System-on-Chip, SoC) has become technically feasible.”)¹; • Ex-1005, 2 (“SoCs represent high-complexity, high-value semiconductor products that incorporate building blocks from multiple sources (either in-house made or externally supplied): in particular, general-purpose fully programmable processors, co-processors, DSPs, dedicated hardware accelerators, memory blocks, I/O blocks, etc. Even though commercial products currently exhibit only a few integrated cores (e.g., a RISC

¹ All bold, italicized emphasis is added unless otherwise noted.

	general purpose MIPS core and a VLIW Trimedia processor for the Philips Nexperia platform for multimedia applications), in the next few years technology will allow the integration of thousands of cores, making a large computational power available.”); • Ex-1005, 3 (“Designing communication architectures for highly integrated deep sub-micron SoCs...”); • Ex-1005, 11-12 (“The individual components of SoCs are inherently heterogeneous with widely varying functionality and communication requirements. The communication infrastructure should optimally match communication patterns among these components. Looking for the most efficient solution can be a (possibly automated) step of a more general design methodology of custom domain-specific NoCs, and Xpipes could be used within this framework as a library of components for the synthesis of the selected NoC configuration. Therefore, for a Xpipes-based design methodology, a tool is needed to automatically instantiate network building blocks (switches, network interfaces and links) from the library of composable soft macros described in SystemC. XpipesCompiler [32] is the tool designed to automatically instantiate a Xpipes-based application-specific NoC for heterogeneous on-chip multi-processors. The complete XpipesCompiler-assisted NoC design flow is depicted in Fig. 9. From the specification of an application, the designer (or a high-level analysis and exploration tool) creates a high-level view of the SoC floorplan, including nodes (with their network interfaces), links and switches. Based on the clock speed target and link routing, the number of pipeline stages for each link is also determined. The information on the network architecture is then specified in an input file for the XpipesCompiler. Routing tables for the network
--	---

	interfaces are also specified. The tool takes the Xpipes library of soft network components as an additional input. The output is a SystemC hierarchical description, which includes all switches, links, network nodes and interfaces and specifies their connectivity. Then the final description can be compiled and simulated at the cycle-accurate and signal-accurate level. At this point, the description can be fed to backend RTL synthesis tools for silicon implementation.”). A POSA would understand that SoCs are a type of integrated circuit. See, for example, Ex-1001, 1:8-13.
--	---

95. [1.pre]/[7.pre]

Claim language	Evidence and disclosure
[1.pre]/[7.pre] An integrated circuit comprising	Bertozzi discloses this element. Bertozzi teaches the use of the Xpipes architecture and platform to design a NoC interconnect for a System-on-Chip (“SoCs”). <i>E.g.</i>, • Ex-1005, Title (“Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip.”); • Ex-1005, 1 (“In this paper, an advanced NoC architecture, called Xpipes, targeting high performance and reliable communication for on-chip multi-processors is introduced.”); 19 (“The integration of an entire system onto the same silicon die (System-on-Chip, SoC) has become technically feasible.”); • Ex-1005, 2 (“SoCs represent high-complexity, high-value semiconductor products that incorporate building blocks from multiple sources (either in-house made or externally supplied): in

	particular, general-purpose fully programmable processors, co-processors, DSPs, dedicated hardware accelerators, memory blocks, I/O blocks, etc. Even though commercial products currently exhibit only a few integrated cores (e.g., a RISC general purpose MIPS core and a VLIW Trimedia processor for the Philips Nexperia platform for multimedia applications), in the next few years technology will allow the integration of thousands of cores, making a large computational power available.”); • Ex-1005, 2 (“<i>The integration of an entire system onto the same silicon die</i> (System-on-Chip, SoC) has become technically feasible.”); • Ex-1005, 3 (“Designing communication architectures for highly integrated deep sub-micron SoCs...”); <i>see also</i> Ex-1005, 28-29. <i>See also</i> [10.pre]. A POSA would understand SoCs are a type of integrated circuit. Ex-1001, 1:8-13.
--	--

96. [10.pre.a]/[1.a]/[7.a]

Claim language	Evidence and disclosure
[10.pre.a]/[1.a]/[7.a] a plurality of functional blocks, and	Bertozzi discloses this element. Bertozzi discloses SoCs with a plurality of cores or building blocks that a POSA would understand to be functional blocks, including (but not limited to) processors, co-processors, CPUs, DSPs, dedicated hardware accelerators, memory blocks, RAM, and I/O blocks. For example: • Ex-1005, 1 (“SoCs represent high-complexity, high-value semiconductor products that incorporate building blocks from multiple sources

(either in-house made or externally supplied): in particular, general-purpose fully programmable processors, co-processors, DSPs, dedicated hardware accelerators, memory blocks, I/O blocks, etc. Even though commercial products currently exhibit only a few integrated cores (e.g., a RISC general purpose MIPS core and a VLIW Trimedia processor for the Philips Nexperia platform for multimedia applications), in the next few years technology will allow the integration of thousands of cores, making a large computational power available.”);

- Ex-1005, Figs. 1, 2, 10, 11.

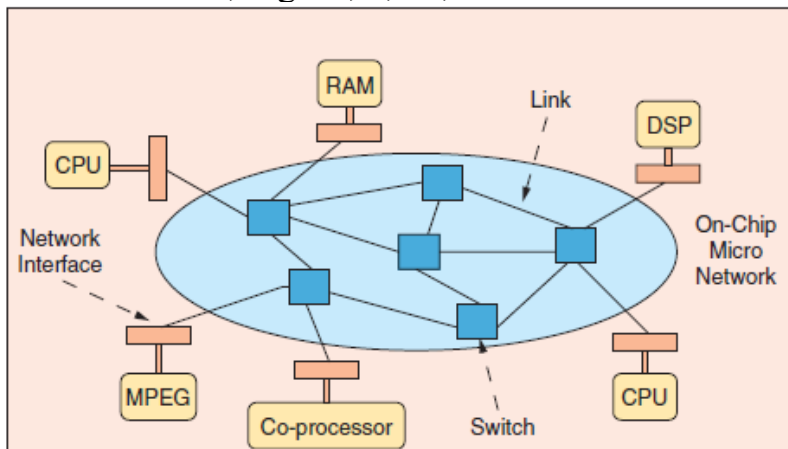


Figure 1. Example of Network-on-Chip architecture.

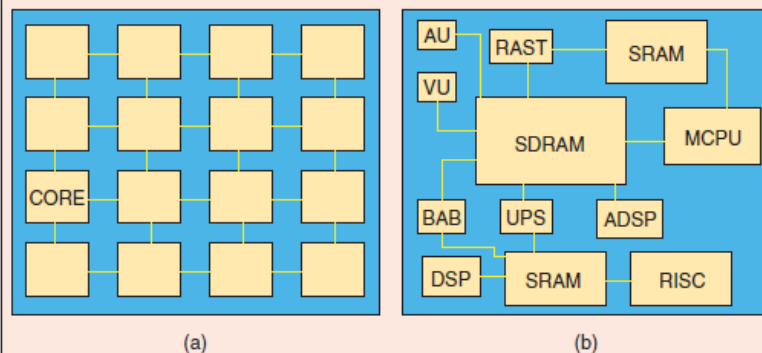
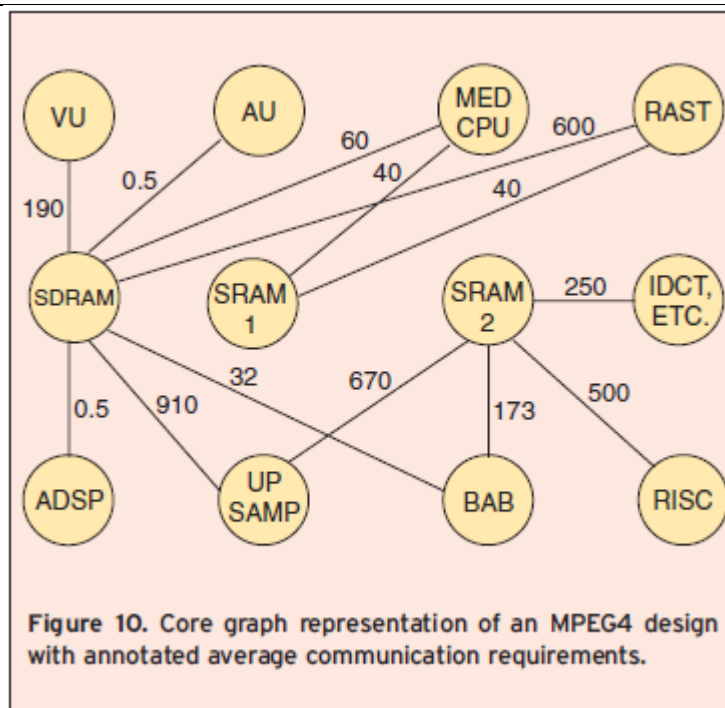
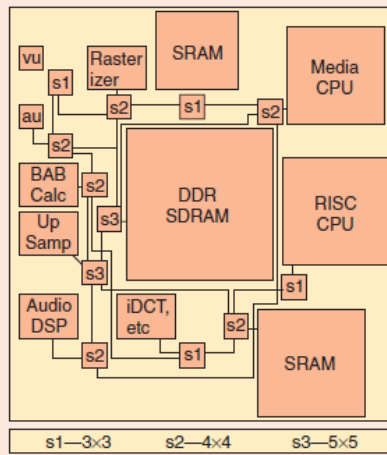
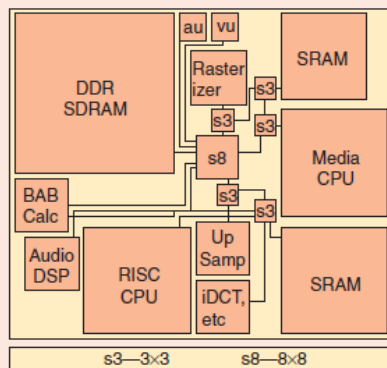


Figure 2. Homogeneous versus heterogeneous architectural template. (a) Tile-Based On-Chip Multi-Processor. (b) MPEG-4 MPSoC.

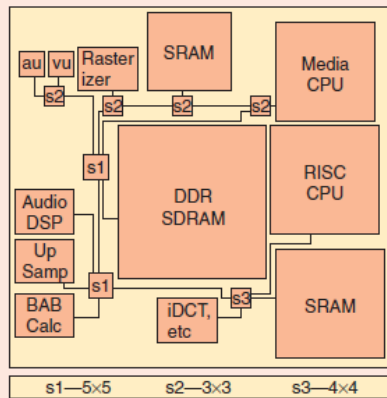




(a)



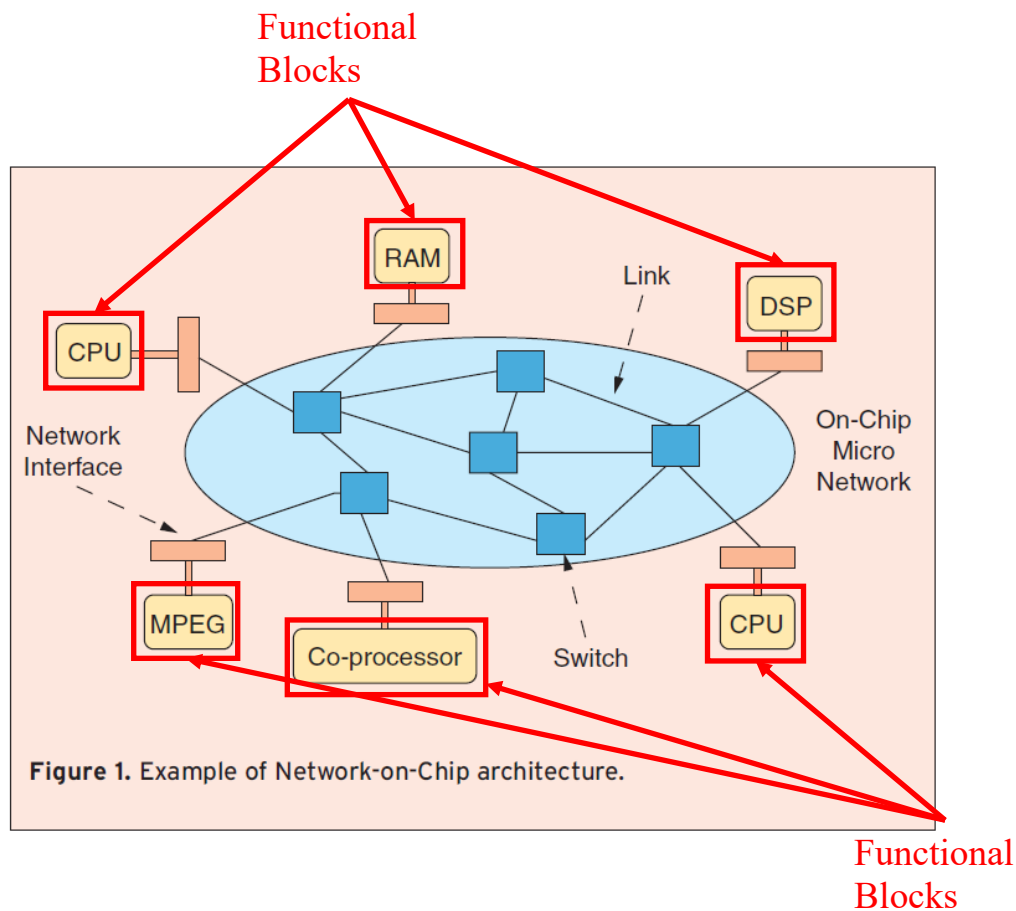
(b)



(c)

Figure 11. NoC configurations for the MPEG4 decoder. (a) MESH NoC. (b) Application-specific NoC No.1. (c) Application-specific NoC No.2.

97. Below I have annotated functional blocks on Fig. 1 of Bertozzi:

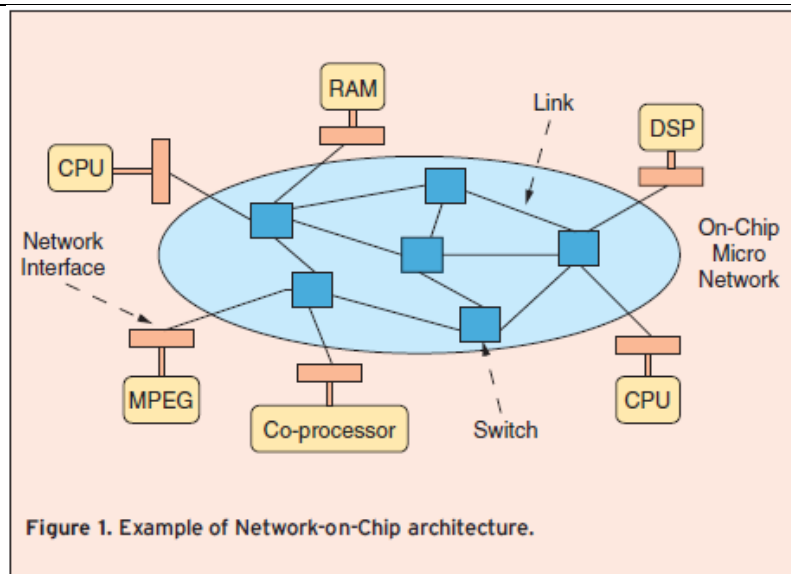


Ex-1005, Fig. 1 (red annotation added). As I explained above in paragraph 75, a POSA would understand this general NoC architecture to be the architecture for Bertozzi's Xpipes.

98. [10.pre.b]/[1.b]/[7.b]

Claim language	Evidence and disclosure
[10.pre.b]/[1.b]/[7.b] a data communication network comprising a plurality of network stations being	This element is disclosed by Bertozzi. A POSA would understand that Bertozzi discloses a Network-on-Chip (" <i>data communication network</i> ") comprising a plurality of network interfaces and routers

interconnected via a plurality of communication channels for communicating data packages between the functional blocks,	or switches (“<i>network stations</i>”) being interconnected via a plurality of links (“<i>communication channels</i>”) for communicating data packets (“<i>data packages</i>”) between Bertozzi’s cores (“<i>functional blocks</i>”). For example: • Ex-1005, 1 (“<i>an on-chip packet-switched micro-network of interconnects, generally known as Network-on-Chip (NoC)</i> architecture [3, 4, 5]. The basic idea is borrowed from traditional large-scale multi-processors and the wide-area networks domain, <i>and envisions on-chip router (or switch)-based networks on which packetized communication takes place</i>, as depicted in Fig. 1. Cores access the network by means of proper <i>interfaces</i>, and have their packets forwarded to destination through a <i>multi-hop routing path.</i>”), • Ex-1005, 2 (“on-chip router (or switch)-based networks on which packetized communication takes place”), • Ex-1005, 4 (“<i>The task of the switch is to carry packets injected into the network to their final destination</i>, following a statically defined or dynamically determined routing path. The switch transfers packets from one of its input ports to one or more of its output ports.”), • Ex-1005, 4-5 (“Data packetization is a critical task for the network interface . . . [m]essages that have to be transmitted across the network are usually partitioned into fixed-length packets”), • Ex-1005, Fig. 1:
--	--



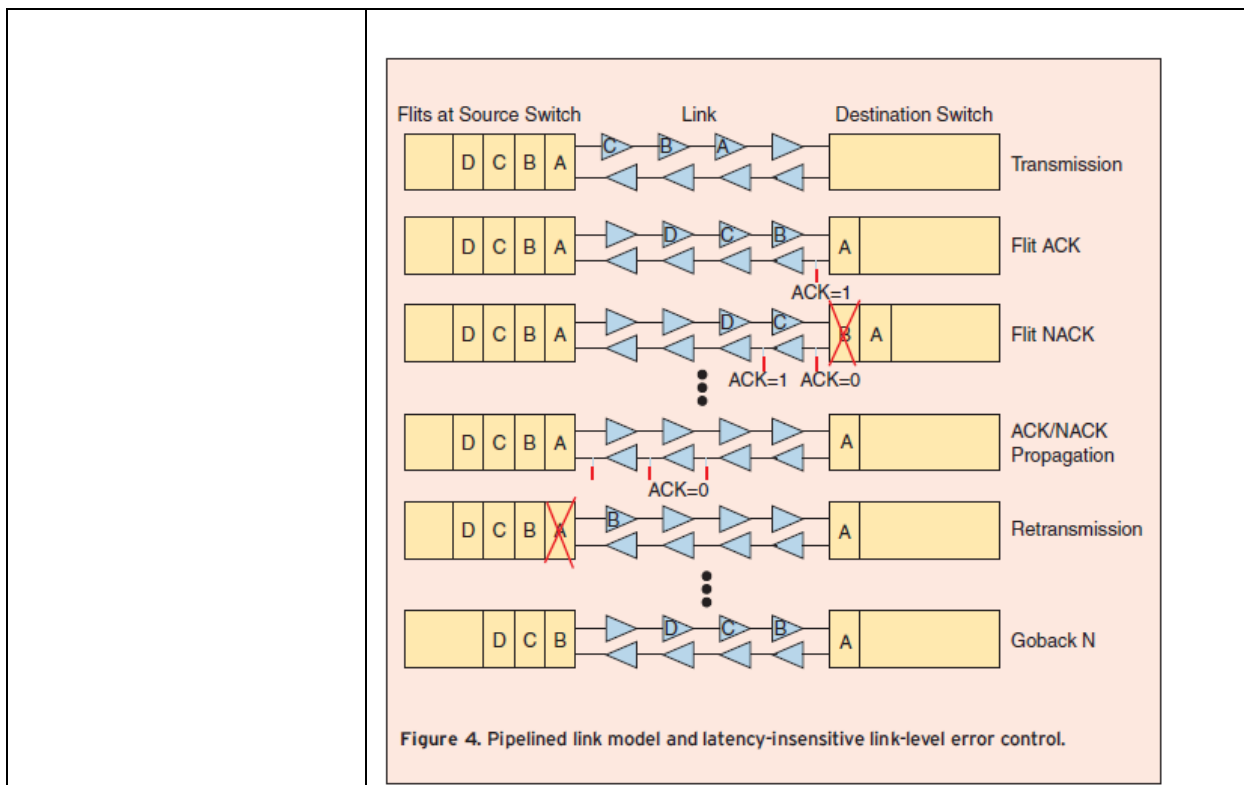
A POSA would understand that Bertozzi’s NoC comprises a ***plurality of network stations*** such as routers (or switches) and network interfaces. For example:

- Ex-1005, 1 (“***envisions on-chip router (or switch)-based networks on which packetized communication takes place***, as depicted in Fig. 1. Cores access the network by means of proper ***interfaces***, and have their packets forwarded to destination through a ***multi-hop routing path***.”);
- Ex-1005, 2 (“Xpipes has been designed as a library of highly parameterised [sic] soft macros (***network interface, switch and switch-to-switch link***)”);
- Ex-1005, 4 (“task of the switch is to carry packets injected into the network to their final destination...”);
- Ex-1005, Fig. 1.

A POSA would also understand that network interfaces and routers are examples of network stations. See Ex-1001, 12:29-31 (“the plurality of network stations comprising a ***plurality of data routers and a plurality of network interfaces***”); 4:31-33 (“***The data routers 110, 120, 130, 140 and the network interfaces 105-108*** comprise the network stations of the data communication

	network 100.”), 5:9-10 (“network stations, e.g. data routers 120 and 140”); 5:51-54 (“in accordance with the delay characteristics of the communication channel 150 between two network stations, e.g. data routers 120 and 140”). A POSA would understand that the routers (or switches) and network interfaces are interconnected via a plurality of links, and a POSA would understand that such links were <u>communication channels</u>. For example: • Ex-1005, 6 (“Finally, the design of communication (switch-to-switch and network interface-to-switch) links represents a critical issue with respect to the system performance. As geometries shrink, gate delay improves much faster than the delay in long wires. Therefore, the long wires increasingly determine the maximum clock rate, and hence performance, of the entire design. The problem becomes particularly serious for domain-specific heterogeneous SoCs, where the wire structure is highly irregular and may include both short and extremely long switch-to-switch links. Techniques have to be devised to decouple link throughput from its length and to allow functionally correct operation of the switches in presence of mis-aligned inputs due to the unequal length of the input links.”); • Ex-1005, 7 (“<i>Xpipes</i> interconnect makes use of pipelined links and of latency-insensitive operation of its building blocks. Switch-to-switch links are subdivided into basic segments whose length can be tailored to the desired clock frequency that needs to be achieved in the design.”); • Ex-1005, Fig. 1 (above). Bertozzi’s routers (or switches), network interfaces, and links are for communicating <u>data packages</u> between the cores. For example:
--	--

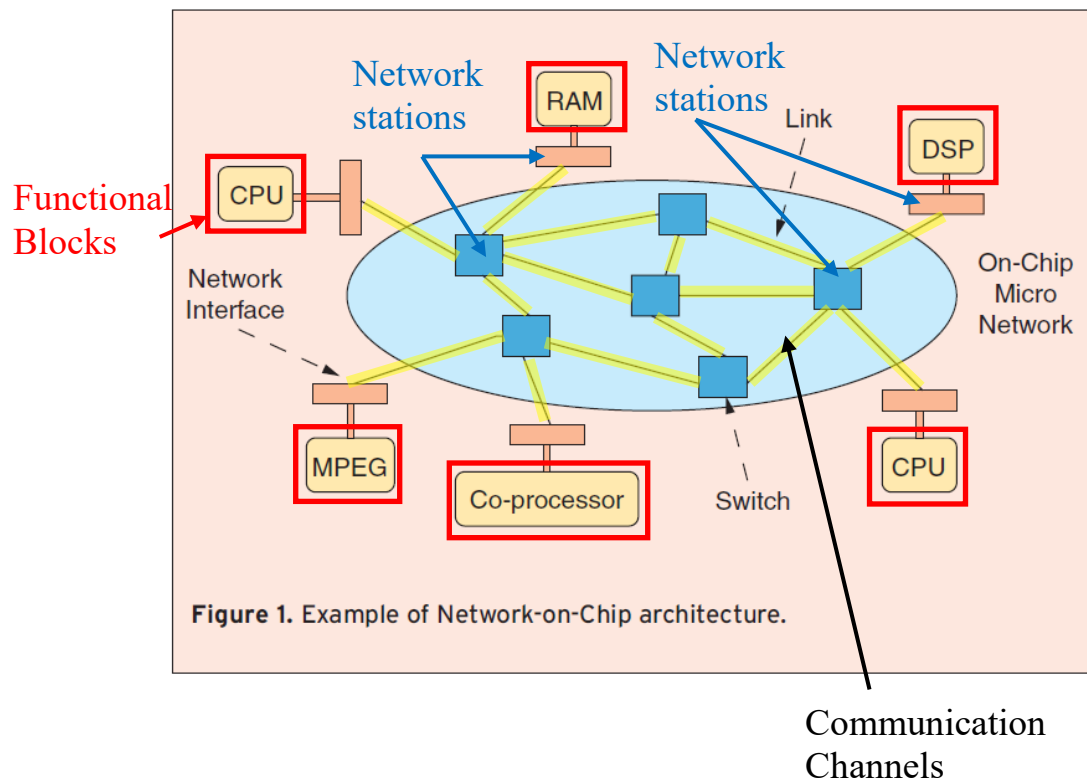
	• Bertozzi teaches that “Messages that have to be <i>transmitted across the network</i> are usually partitioned into <i>fixed-length packets</i>.” Ex-1005, 5; • Bertozzi further teaches that “<i>Data packetization is a critical task for the network interface....</i>The packet preparation process consists of building the packet header, payload and packet tail.” ... “The switch transfers packets from one of its input ports to one or more of its output ports.” Ex-1005, 5. • Ex-1005, 5 (“<i>The task of the switch is to carry packets injected into the network to their final destination</i>, following a statically defined or dynamically determined routing path. The switch transfers packets from one of its input ports to one or more of its output ports.”) • Bertozzi explains that “The <i>packet preparation process</i> consists of building the <i>packet header, payload and packet tail</i>. The <i>header contains the necessary routing and network control information</i> (e.g., the source and destination address). When source routing is used, the destination address is ignored and replaced with a route field that specifies the route to destination. This overhead in terms of packet header is counterbalanced by a simpler routing logic at the network switches: they simply have to look at the routing fields in the packets and route them over the specified switch output port. The packet tail indicates the end of a packet and usually contains parity bits for error-detecting or error-correcting codes. Packetizing data deals effectively with communication errors. In fact, packet boundaries contain the effect of errors and allow error recovery on a packet-by-packet basis.” Ex-1005, 5. A POSA would understand this description to apply to all embodiments of the system Bertozzi describes designing. • Ex-1005, Fig. 1 (above), Figs. 3, 4
--	--



99. A POSA would understand that Bertozzi’s Network-on-Chip (“NoC”) was a *data communications network*. Bertozzi discloses that its NoC is “an on-chip packet-switched micro-network of interconnects” that communicates data packets between Bertozzi’s cores such as the “CPUs,” “RAM,” “MPEG,” “Co-processor,” and “DSP.” Ex-1005, Title, Abstract, 19; Figs. 1, 3, 4. Bertozzi discloses that the data packets that are communicated between its “cores” each carry multiple data elements. *E.g.*, Ex-1005, 3, 5-6 and Fig. 4 (embodiment showing data packet with four data elements A, B, C, D). Bertozzi’s packets are data packages as they are made up of data elements. The patent explains that a data

package contains N data elements. *E.g.*, Ex-1001, 2:3-5; 2:17-22. Thus Bertozzi teaches a network for communicating data packages as claimed.

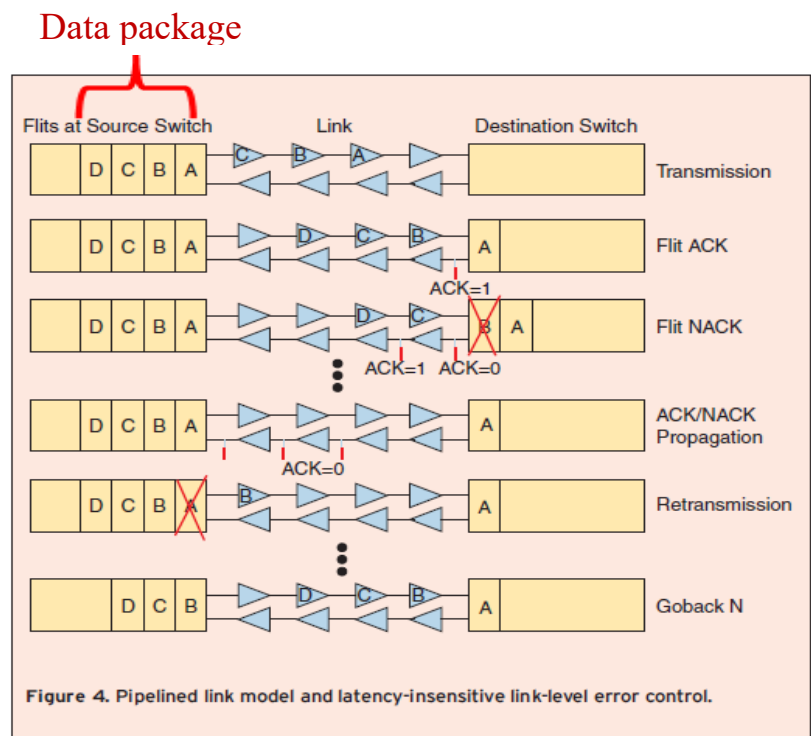
100. Below I have annotated the network stations and communication channels on Fig. 1:



Ex-1005, Fig. 1 (annotated), 3.

101. A POSA would recognize that the data packets disclosed by Bertozzi would be the claimed *data package*. The '2893 patent teaches that “data packages have a fixed size of N data elements.” Ex-1001, 2:18. Similarly, Bertozzi teaches that the data packets that are communicated between its “cores” each carry multiple data elements. Ex-1005, e.g., 5 (“Packets in turn are often broken into

message flow control units called flits.”), 7-8; *see also infra* [10.pre.c]. Figure 4 shows an embodiment with four data elements A, B, C, D. I have reproduced it below:



Ex-1005, Fig. 4 (annotated).

102. [10.pre.c]/[1.c]/[7.c]

Claim language	Evidence and disclosure
[10.pre.c]/[1.c]/[7.c] each data package comprising N data elements including a data element comprising routing information for the network stations, N	This element is disclosed by Bertozzi under both Petitioner Qualcomm’s construction and Patent Owner’s construction. Specifically, Bertozzi discloses data packets (“<i>each data package</i>”) comprising N flits (“<i>comprising N data elements</i>”) including a flit (“<i>data element</i>”) comprising routing information of the network interfaces and routers

being an integer of at least two,	(“<i>routing information for the network stations</i>”), N being an integer of at least two. I explain this further in the three subsections below.
-----------------------------------	---

1. “each data package comprising N ***data elements*** . . .”

103. Bertozzi discloses that its packets can be “partitioned into fixed-length packets” (“*each data package comprising N data elements*”) and broken down into N number of “data elements,” which Bertozzi are referred to as “flits”:

Data packetization is a critical task for the network interface, and has an impact on the communication latency, in addition to the latency of the communication channel. Messages that have to be transmitted across the network are usually ***partitioned into fixed-length packets***. Packets in turn are often ***broken into message flow control units called flits***. In the presence of channel width constraints, multiple physical channel cycles can be used to transfer a single flit. A phit is the unit of information that can be transferred across a physical channel in a single step. Flits represent logical units of information, as opposed to phits that correspond to physical quantities. In many implementations, a ***flit is set to be equal to a phit***.

Ex-1005, 5. A POSA would understand both the teaching: (1) that messages can be partitioned into “fixed length” packets and (2) that packets are often broken into message flow control units called flits to apply to all of Bertozzi’s embodiments

104. A POSA would understand that in Bertozzi, each flit of a fixed-length packet corresponds to the amount of data sent over one clock cycle and thus to a “*data element*” as claimed by the ’2893 patent, for several reasons, which I explain in the next several paragraphs.

105. I have reproduced Bertozzi’s discussion of flits from page 5 below:

Data packetization is a critical task for the network interface, and has an impact on the communication latency, in addition to the latency of the communication channel. Messages that have to be transmitted across the network are usually partitioned into fixed-length packets. Packets in turn are often broken into message flow control units called *flits*. In the presence of channel width constraints, multiple physical channel cycles can be used to transfer a single flit. A *phit* is the unit of information that can be transferred across a physical channel in a single step. Flits represent logical units of information, as opposed to phits that correspond to physical quantities. In many implementations, a flit is set to be equal to a phit.

106. The first relevant sentence is: “Packets in turn are often broken into message flow control units called flits.” Ex-1005, 5. It introduces the concept of a flit. The next two sentences introduce the concept of a phit: “In the presence of channel width constraints, multiple physical channel cycles can be used to transfer a single flit. A phit is the unit of information that can be transferred across a physical channel in a single step.” *Id.* From these two sentences, a POSA would understand that a phit is the amount of data sent over a clock cycle, and its size is dictated by the number of wires in the physical channel. This is exactly how the

'2893 patent uses the term "data element." *See, e.g.*, Ex-1001, 2:17-26, 6:31-7:2 ("subsequent data elements d0 and d1 of this data package are received by the data router 120 at respective clock cycles t+1 and t+2"), 7:58-8:26; Figs. 1-4. As a POSA would understand that a phit is the amount of data that is transmitted over a single clock cycle, a POSA would understand a phit to be a data element within the meaning of the '2893 patent.

107. A POSA would also understand that there is a ratio of flits to phits. *Id.* ("multiple physical channel cycles can be used to transfer a single flit....In many implementations, a flit is set to be equal to a phit."). A POSA would understand that in Bertozzi's implementation a flit is set equal to a phit. Ex-1005, 5. Bertozzi discusses flits in the context of being transferred over one clock cycle, which a POSA would understand as requiring a flit to be set equal to a phit, and thus, for a flit in the implementation discussion of Bertozzi to correspond to the amount of data sent over a clock cycle. On page 9, it discusses that it takes a flit N cycles to propagate across a link, where N is the link length, i.e., the number of repeaters on the link. Ex-1005, 9 ("where N is the link length (expressed as number of basic repeater stages)...the flit has to propagate across the link (N cycles).") If a flit were set to anything other than being equal to a phit, it would not take N cycles to propagate across the link. Similarly, Bertozzi discusses flits propagating through the network in terms of cycles. Ex-1005, 7 ("According to the link length, a

certain number of clock cycles is needed by a flit to cross the interconnect.”), 8
(“When flits are correctly received at the destination switch, an ACK is propagated back to the source, and after *N clock cycles (where N is the link length expressed in terms of number of repeater stages)* the flit will be discarded from the buffer of the source switch.”).

108. Earlier articles regarding the Xpipes system confirm that a flit is sent over “one clock cycle.” Ex-1010, 2 (“In our NoC design, a single flit can be pushed into the network in one clock cycle.”).

109. Bertozzi also shows in Figure 4 flits being stepped across a channel one clock cycle at a time. Ex-1005, 7-8, Fig. 4. This is exactly how the ’2893 patent specification uses the term “data element.” *E.g.*, Ex-1001, 6:31-7:2 (*e.g.*, “subsequent data elements d0 and d1 of this data package are received by the data router 120 at respective clock cycles $t+1$ and $t+2$.”).

110. Because in Bertozzi’s system a “flit” is set equal to a “phit” such that a “flit” is set equal to the amount of data sent over one clock cycle and corresponds to the claimed “data element”, I have used the term “flit” throughout the remainder of declaration. The remaining implementation disclosure of Bertozzi discusses flits and uses it as the implementation unit.

111. I note that the word “flits” also appears once in the ’2893 specification, which notes that data packages are “sometimes” referred to as flits.

Ex-1001, 4:56-57 (“Such data packages are sometimes referred to as flow instruction units (flits).”). However, as I discussed above, depending on the implementation a “flit” can have one data element or multiple data elements. In Bertozzi, the flit is set to contain one data element (i.e. a ratio of 1:1), and therefore a flit in Bertozzi’s implementation corresponds to a data element. Ex-1005, 5. In the ’2893 patent, a flit is set to contain multiple data elements (e.g., a flit-to-data element ratio of 1:2, or 1:3, etc.); therefore, in that implementation a flit corresponds to a data package, rather than a data element.

2. “including a data element comprising *routing information for the network stations*. . .”

112. Bertozzi discloses that each of Bertozzi’s packets have a header flit (*a data element*) that contains “the **necessary routing and network control information**” (“routing information for the network stations.”). Ex-1005, 5. The header is the first flit of a packet. Ex-1005, 5 (“The *first flit of a packet contains routing information and header flit* decoding enables the switches to *establish the path* and subsequent flits *simply follow this path in a pipelined fashion* by means of switch output port reservation.”). A POSA would understand that the “necessary routing and network control information” is the claimed “routing information” for the network stations.

113. For example, Bertozzi explains:

The packet preparation process consists of building the packet header, payload and packet tail. *The header contains the necessary routing and network control information (e.g., the source and destination address).* When source routing is used, the destination address is ignored and replaced with a route field that specifies the route to destination. This overhead in terms of packet header is counterbalanced by a simpler routing logic at the network switches: they simply have to look at the routing fields in the packets and route them over the specified switch output port. The packet tail indicates the end of a packet and usually contains parity bits for error-detecting or error-correcting codes. Packetizing data deals effectively with communication errors. In fact, packet boundaries contain the effect of errors and allow error recovery on a packet-by-packet basis.

Ex-1005, 5.

3. “data package **comprising** N data elements . . . N being an integer of at least two”

114. Bertozzi teaches a “*data package comprising N data elements . . . N being an integer of at least two*” under both Petitioner Qualcomm’s and Patent Owner’s constructions.

Under Petitioner Qualcomm’s Construction

115. Petitioner Qualcomm's Construction requires that "every data package communicated by the network stations has a fixed size of N data elements." Ex-1025, 32. This is disclosed by Bertozzi. *See* Ex-1005, 5-7; Fig. 4.

116. Bertozzi teaches that messages "transmitted across that network are *usually partitioned into fixed-length packets*." Ex-1005, 5. A POSA would understand that Bertozzi means that each data packet is the same size or length.² This is the same meaning that a POSA would prescribe to the '2893 patent's use of the term "fixed size" when discussing its data packages. *See* Ex-1001, 2:18 ("data packages have a fixed size of N data elements"). A POSA would have understood

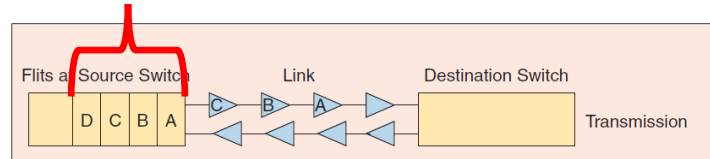
² As I noted above regarding Fixed-Length Packets, a POSA would understand that fixed-length packets means that each data packet is the same size or length. *See* Ex-1015, 3 ("ATM packets are fixed-sized packets with data fields of 48 bytes. Five bytes are used for header, so the length of the whole packet is 53 bytes."); *see also* Ex-1035, 4 ("fixed length: Pertaining to a record of field that has a constant length, regardless of the specific data contained in it. Filler characters may be used to maintain the fixed length. *Contrast*: variable length. *See also*: fixed format.") and 3 (fixed format: A file organization in which all logical records in the file of fixed length. *Synonym*: F format. *Contrast*: variable format.") (emphasis in original).

that in a network with messages partitioned into fixed-length packets, all of Bertozzi's "fixed-length packets" would have the same fixed length meaning a certain number of flits. Ex-1005, 5 ("Packets in turn are often broken into messages flow control units called flits."). As discussed above, in Bertozzi, each flit is a "*data element*" as claimed in the '2893 patent.

117. A POSA would understand that Bertozzi teaches a method of designing an integrated circuit that includes using fixed-length packets. A POSA would know that fixed-length packets are an option within the Xpipes platform. Alternatively, a POSA would at once envisage an implementation of a NoC using the Xpipes platform where the packets were fixed-length and thus every data package communicated by the network stations has a fixed size of N data elements. A POSA would understand that there are various advantages associated with using fixed-length packets, including simplified routers, error detection, error correcting, and timing across the NoC.

118. In view of Bertozzi's teachings, the number of flits ("data elements") in Bertozzi's packet discloses the claim "each data package comprising N data elements..., N being an integer of at least two." In Figure 4, Bertozzi shows a packet embodiment ("data package") that has a fixed length of N=4 data elements (in Bertozzi's implementation also called flits) lettered A through D:

data package with a fixed size of N=4 data elements



Ex-1005, 7-8.

119. As I explained previously, a POSA would understand Figure 4 to be showing a packet with four data elements. The flits (“data elements”) shown in Figure 4 are lettered A through D and are stored within a switch, the switch shows no additional data elements following, and shows a blank space after the last data element. Ex-1005, Fig. 4, p.6 (Xpipes is a “packet-switched NoC[]”). Thus, a POSA looking at Figure 4 would understand the flits lettered A, B, C, D to be a data packet. In this example, $N=4$, an integer of at least two.

120. A POSA would have understood that Bertozzi discloses designing its integrated circuit embodiments for use with fixed length packets. Ex-1005, 5 (“Messages that have to be transmitted across the network are *usually partitioned into fixed-length packets.*”). In a network with messages partitioned into fixed-length packets, a POSA would have understood that all of the packets have the same fixed length (four flits in the Fig. 4 embodiment described above). Ex, 1005, 5, 7-8, Fig. 4. A POSA would further understand that the length of these fixed-length packets does not change as the packet is transmitted through the network. Ex-1005, Fig. 4. As such, Bertozzi discloses this element under both Petitioner

Qualcomm's and the Patent Owner's proposed construction. In the event that Patent Owner incorrectly argues that the four flits shown in Fig. 4 are not a packet, a POSA would in any event at once envisage that the example of fixed-length packets taught by Bertozzi corresponds to a data package having four flits A, B, C, D (each one a data element) as shown in Figure 4.

121. As discussed above, each packet ("data package") includes a flit ("data element") comprising routing information for the network stations. Ex-1005, 5. Packets built in an Xpipes NoC have a packet header that contains "***routing and network control information (e.g., the source and destination address)***." Ex-1005, 5. The header is the first flit of a packet. Ex-1005, 5 ("The first flit of a packet contains routing information"); *see also* Ex-1005, 10-11.³

Under Patent Owner's Construction

³ As I have previously discussed, Bertozzi also shows an Xpipes embodiment for a data packet with ten flits, at Figure 3. A POSA would understand that this is a general illustration of the partitioning procedure of a packet ("***Data packetization results in the packet partitioning procedure illustrated in Fig. 3,***" Ex-1005, 6) and is a different embodiment than the specific embodiment from the four-flit data packet embodiment shown at in Figure 4 that represent transmission between two consecutive switches.

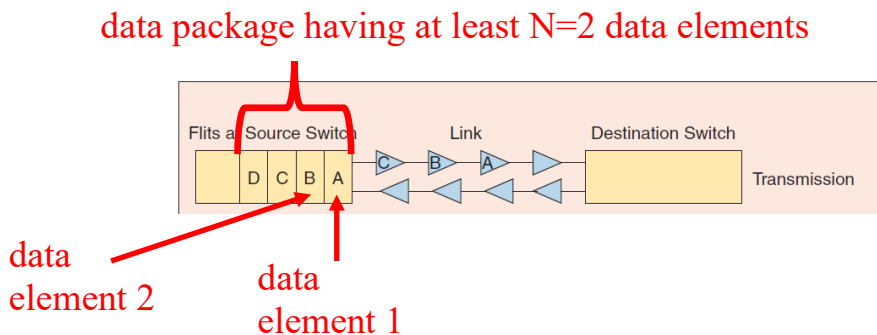
122. Under Patent Owner's construct, Bertozzi discloses this element [10.pre.c] under Patent Owner's construction, in multiple ways. One, Bertozzi discloses this element is met by Xpipes' fixed-length packet embodiment described above for the same reasons as under Petitioner Qualcomm's narrower construction as discussed above. Two, under Patent Owner's interpretation that the claim only requires each data package to have "at least N data elements" and each data package does not need to be the same size. *Supra*, §VIII. Thus, Bertozzi disclose this element as it explains that its each data packet ("data package") has multiple flits ("data elements"). Bertozzi describes that each packet includes at least a packet header flit and a separate packet tail flit, such that each contains at least N=2 "data elements." Each of the packets as taught by Bertozzi has at least a header flit and a tail flit, meaning that each packet ("data package") would have at least two flits such that N=2. Ex-1005, 5:

The packet preparation process consists of ***building the packet header, payload and packet tail***. The ***header contains the necessary routing and network control information*** (e.g., the source and destination address). When source routing is used, the destination address is ignored and replaced with a route field that specifies the route to destination. This overhead in terms of packet header is counterbalanced by a simpler routing logic at the network switches: they simply have to look at the routing fields in the packets and route them over the specified switch

output port. The *packet tail indicates the end of a packet* and usually contains parity bits for error-detecting or error-correcting codes. Packetizing data deals effectively with communication errors. In fact, packet boundaries contain the effect of errors and allow error recovery on a packet-by-packet basis.

123. Each flit has a “flit type field” that identifies whether the flit is a header, tail, or payload. Ex-1005, 6 (“A *flit type field allows to identify the head and the tail flit*, and to distinguish between header and payload flits.”). As packet headers and packet tails have separate flit type fields, they must be on independent flits. Further, the number of flits remains constant once the packet is injected into the network by the NIU.

124. For example, Figure 4 shows at least two data elements:



Ex-1005, Fig. 4 (showing, e.g., “A”, “B”) (red annotations added).

125. As discussed above, Bertozzi discloses that each data packet contains routing information in its header. *Supra*, ¶¶112, 121.

126. Bertozzi's Figure 3 shows a different embodiment where packets have ten flits. Ex-1005, Fig. 3. Under Patent Owner's broader construction, claim 10 would have also read on this different embodiment. Under Patent Owner's construction, one only needs to point to at least two data elements in a data package, even if there are more than two data elements in a package. So under Patent Owner's construction, Figure 3 shows a data package with (at least) N=2 data elements, one of which comprises routing information for the network stations. For example, see Ex-1005, 5 (header contains "routing information").

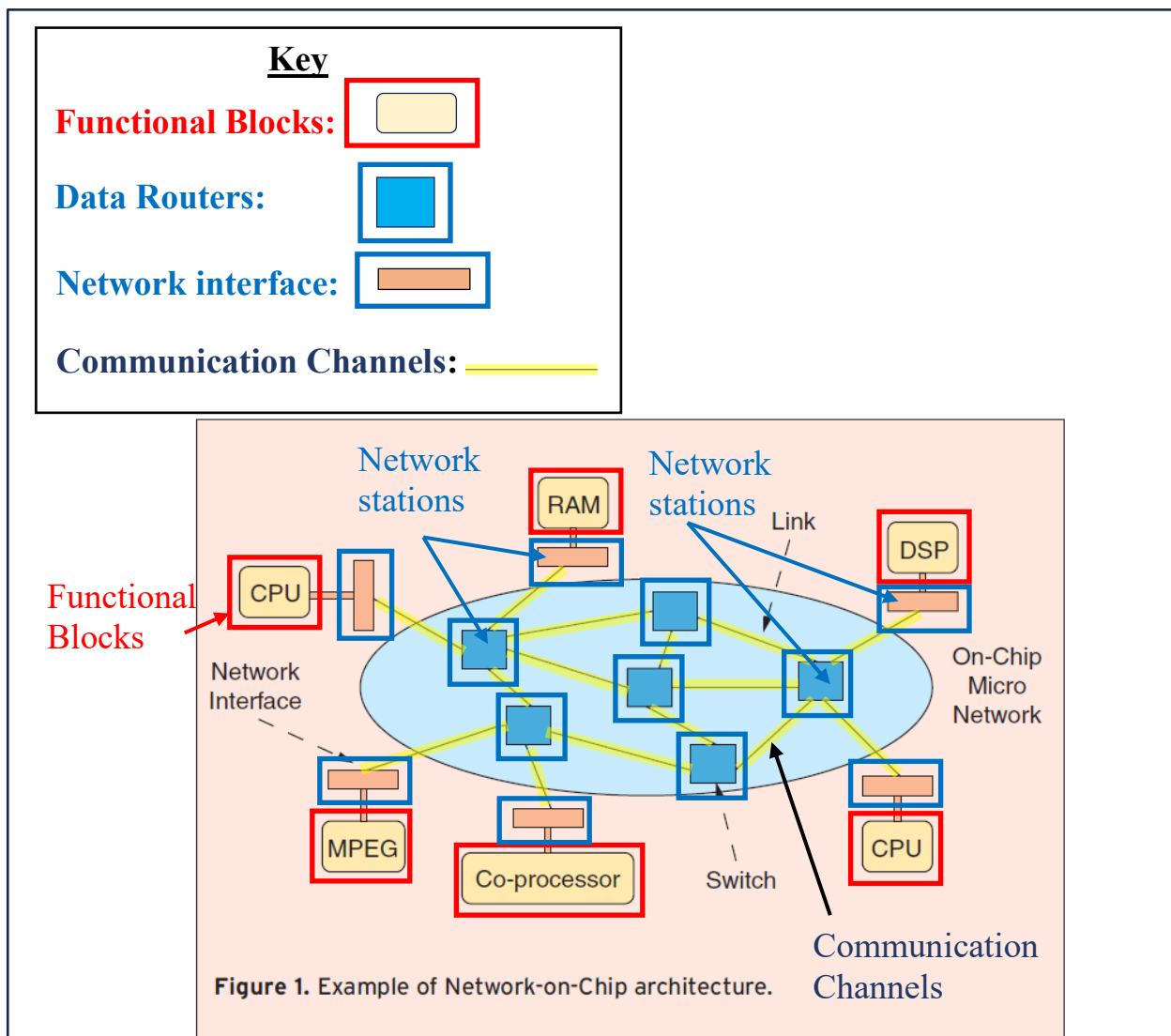
127. [10.pre.d]/[1.d]/[7.d]

Claim language	Evidence and disclosure
[10.pre.d]/[1.d]/[7.d] the plurality of network stations comprising a plurality of data routers and a plurality of network interfaces, each of the data routers being coupled to a functional block via a network interface;	This element is disclosed by Bertozzi. Bertozzi discloses that the plurality of network stations comprises a plurality of switches or routers (<i>data routers</i>) and a plurality of network interfaces (<i>network interfaces</i>), each of the switches or routers (<i>data routers</i>) being coupled to a core (<i>functional block</i>) via a network interface (<i>network interface</i>). For example: Ex-1005, 2 ("A scalable communication infrastructure that better supports the trend of SoC integration consists of <i>an on-chip packet-switched micro-network of interconnects, generally known as Network-on-Chip (NoC)</i> architecture [3, 4, 5]. The basic idea is borrowed from traditional large-scale multi-processors and the wide-area networks domain, and <i>envisions on-chip router (or switch)-based networks</i> on which <i>packetized communication takes place</i>, as depicted in Fig. 1. Cores access the network by means of proper

	<i>interfaces</i>, and have their packets forwarded to destination through a <i>multihop routing path</i>.”), • Ex-1005, 3 (“Xpipes has been designed as a library of highly parameterised [sic] soft macros (<i>network interface, switch and switch-to-switch link</i>), which are design-time tunable and composable.”), • Ex-1005, 4-5 (“In general, the most relevant tasks of a network interface (NI) are: (i) <i>hiding the details about network communication protocol to the cores</i>, so that they can be developed independently of the communication infrastructure, (ii) communication protocol conversion (from end-to-end to network protocol), (iii) <i>data packetization</i> (packet assembly, delivery and disassembly).”); • Ex-1005, 5 (“The task of the <i>switch is to carry packets injected into the network to their final destination....</i>”); • Ex-1005, Fig. 1. A POSA would understand that the plurality of routers or switches and network interfaces disclosed by Bertozzi are the claimed <i>plurality of network stations</i>. See Ex-1001, 12:29-31 (“the plurality of network stations comprising a plurality of data routers and a plurality of network interfaces”); 4:31-33 (“The data routers 110, 120, 130, 140 and the network interfaces 105-108 comprise the network stations of the data communication network 100.”), 5:9-10 (“network stations, e.g. data routers 120 and 140”); 5:51-54 (“in accordance with the delay characteristics of the communication channel 150 between two network stations, e.g. data routers 120 and 140”). Each of the switches or routers (“data routers”) is coupled to a core (“functional block”) via a network interface (“network interface”). E.g., Ex-1005, Fig. 1. The switches or routers “carry packets injected into the
--	---

	network to their final destination.” Ex-1005, 5. The network interfaces interact directly with the cores and create packets to be injected into the network from the messages communicated by the cores. For example; • Ex-1005, 6 (“Xpipes network interfaces use OCP as point-to-point communication protocol, and takes care of protocol conversion to adapt to the network protocol.”); • Ex-1005, 5 (“Data packetization is a critical task for the network interface”); • Ex-1005, 10 (“The Xpipes network interface provides a standardized OCP interface to networked cores. The NI for cores that initiate communication (initiators) needs to turn OCP-compliant transactions into packets to be transmitted across the network. It represents the slave side of an OCP end-to-end connection (the master side being the initiator core), and it is therefore referred to as network interface slave (NIS).”).
--	---

128. I have annotated the functional blocks, links, routers and network interfaces below on Fig. 1:



Ex-1005, Fig. 4 (annotations added). As I have annotated above in Fig. 1, the network on chip disclosed by Bertozzi contains a plurality of data routers/switches and a plurality of network interfaces. The data routers/switches are coupled to the cores via the network interfaces allowing the cores to interact with the network.

Ex-1005, 2.

129. Bertozzi discloses a NoC containing a plurality of routers. Ex-1005, 2 (“The basic idea...*envisioned on-chip router (or switch)-based networks*”), Fig. 1.

While Figure 1's labels say "Switch" rather than data router, Bertozzi uses these terms interchangeably. Ex-1005, 2 (describing "on-chip router (or switch)-based networks"); 5 (describing "routing logic at the network switches"). In any event, a POSA would also at once envisage using data routers where Bertozzi Fig. 1 illustrates "switch[es]". Ex-1005, 2 ("the basic idea [of NoC systems] . . . envisions on-chip router (or switch)-base networks on which packetized communication takes place").

130. The NoC also contains a plurality of network interfaces. *E.g.*, Ex-1005, 3 ("Xpipes has been designed as a library of highly parameterized soft macros ("...network interface..."), 6 ("Xpipes ***network interfaces use OCP as point-to-point communication protocol with the cores***"). The data routers are coupled to the cores via the "network interfaces" allowing the cores to interact with the network. Ex-1005, 2. The ***network interfaces convert between the bus protocol signals communicated by the cores*** (e.g., VCI, OCP) and the packets communicated over the network of "on-chip router[s] (or switch[es])." Ex-1005, 2, 5, 10; Fig. 7. The various network interfaces and networks stations are connected by links made of wires over which packets are transmitted. Ex-1005, 7, Fig. 1.

131. [10.e]

Claim language	Evidence and disclosure
[10.e] the method comprising the acts	This element is disclosed by Bertozzi.

of: identifying a first communication channel between a first network station and a second network station that has a data transfer delay exceeding a predefined delay threshold; and	Bertozzi teaches that Xpipes identifies a link (“<i>first communication channel</i>”) between a source switch (“<i>first network station</i>”) and a destination switch (“<i>second network station</i>”) that has a data transfer delay exceeding a predefined delay threshold. For example: • Ex-1005, 7 (“A solution to overcome the interconnect-delay problem consists of <i>pipelining interconnects</i> [34, 35]. Wires can be partitioned into segments (or <i>relay stations, which have a function similar to the one of latches in a pipelined data path</i>) whose length satisfies <i>pre-defined timing requirements (e.g., the desired clock speed of the design)</i>....Switch-to-switch links are subdivided into basic segments whose length can be tailored to the <i>desired clock frequency</i> that needs to be achieved in the design. In this way, the system operating frequency is not bound by the delay of long links. According to the link length, a certain number of clock cycles is needed by a flit to cross the interconnect.”), • Ex-1005, 8 (“Fig. 4 illustrates the link model, which is equivalent to <i>a pipelined shift register</i>. Pipelining has been used both for data and control lines. The figure also illustrates how pipelined links affect the operation of latency-insensitive link-level error control: the retransmission of a corrupted flit between two successive switches is represented. Multiple outstanding flits propagate across the link during the same clock cycle. When flits are correctly received at the destination switch, an ACK is propagated back to the source, and after N clock cycles (where N is the link length expressed in terms of number of repeater stages) the flit will be discarded from the buffer of the source switch. On the contrary, a corrupted flit is NACKed and will be retransmitted in due time.”);
--	---

- Ex-1005, 12 (“The complete XpipesCompiler-assisted NoC design flow is depicted in Fig. 9. From the specification of an application, the designer (or a high-level analysis and exploration tool) creates a high-level view of the SoC floorplan, including nodes (with their network interfaces), links and switches. **Based on the clock speed target** and link routing, the number of pipeline stages for each link is also determined.”);
- Ex-1005, Fig. 4.

Bertozzi recognizes that “the design of communication (switch-to-switch and network interface-to-switch) links represents a critical issue with respect to the system performance,” and “[t]herefore, long wires increasingly **determine the maximum clock rate**, and hence performance, of the entire design.” Ex-1005, 6. To solve what Bertozzi refers to as the “interconnect-delay problem,” Bertozzi teaches to take long wires, i.e. wires with a delay that is incompatible with the desired clock speed, and break them into “segments whose length can be tailored to the **desired clock frequency** that needs to be achieved in the design.” Ex-1005, 7. Bertozzi teaches that the “[b]ased on the clock speed target and link routing, the number of pipeline stages for each link is also determined. The information on the network architecture is then specified in an input file for the XpipesCompiler.” Ex-1005, 12.

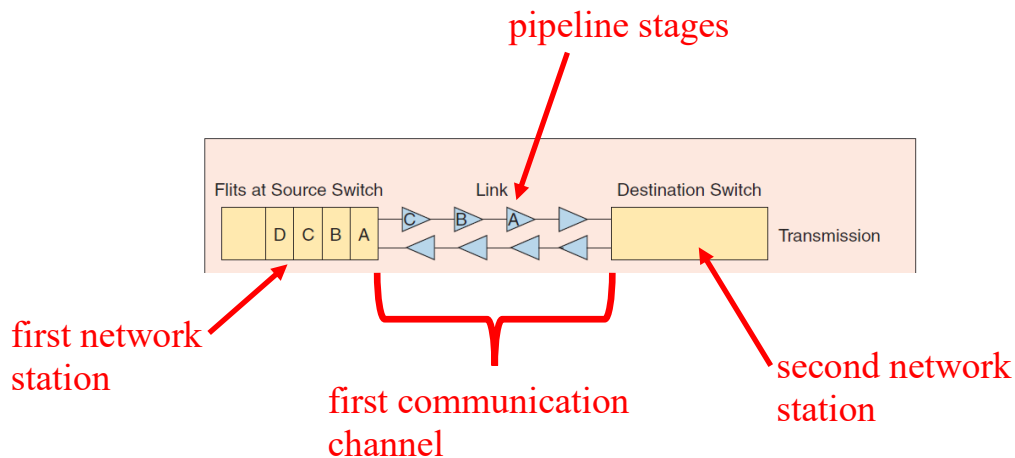
Bertozzi discloses that the Xpipes system is able to identify the “link delay” of wires “whose length satisfies **pre-defined timing requirements (e.g., the desired clock speed of the design)**,” and “[s]witch-to-switch links are subdivided into basic segments whose length can be tailored to the **desired clock frequency** that needs to be achieved in the design.” Ex-1005, 7. Therefore, a POSA would understand that in order to compare a “link delay” (the claim’s “data transfer delay”) to a desired clock rate (i.e., frequency) in order to determine if a length does or does not satisfy a predefined timing requirement, the

	comparison has to be made to a delay threshold that represents the desired clock frequency⁴ of the design. As a result of that comparison, wires are partitioned via repeaters (e.g., relay stations or latches) in order to satisfy the timing requirement. Therefore, Bertozzi describes identifying links with a “link delay” that exceeds “pre-defined timing requirements (e.g., the desired clock speed of the design).” In the Bertozzi section entitled “Network Link,” Bertozzi discloses “A solution to overcome the <i>interconnect-delay problem consists of pipelining interconnects</i> [34, 35]. Wires can be <i>partitioned into segments (or relay stations, which have a function similar to the one of latches in a pipelined data path)</i> whose <i>length satisfies pre-defined timing requirements</i> (e.g., the <i>desired clock speed of the design</i>). In this way, <i>link delay</i> is changed into latency, but the data introduction rate is not bound by the <i>link delay</i> any more. Now the latency of <i>a channel connecting two modules</i> may end up being more than one clock cycle. This requires the system to be composed of modules whose behavior does not depend on the latency of input communication channels [34].” Ex-1005, 7. “<i>Xpipes interconnect makes use of pipelined links</i> and of latency-insensitive operation of its building blocks. <i>Switch-to-switch links are subdivided into basic segments whose length can be tailored to the desired clock frequency</i> that needs to be achieved in the design. In this way, the system operating frequency is not bound by the <i>delay of long links</i>. According to <i>the link length</i>,
--	--

⁴ A POSA would understand that a desired clock rate is the inverse of a desired period (i.e., time elapsed) ($f=1/T$, where f is the clock frequency (i.e. clock rate) and T is the period of the clock). Ex-1034, 134 (“Clock frequency in a digital computer or control, the reciprocal of the period of a single cycle, expressed in terms of the number of cycles occurring in one second of time (hertz, kilohertz, or megahertz)”); Ex-1035, 5 (“frequency”: “The number of periods, or specified fraction of periods, per unit time.”).

	<i>a certain number of clock cycles is needed by a flit to cross the interconnect.” Ex-1005, 7-8.</i>
--	---

132. I have annotated the first network station, the second network station, and the first communication channel below on Figure 4:

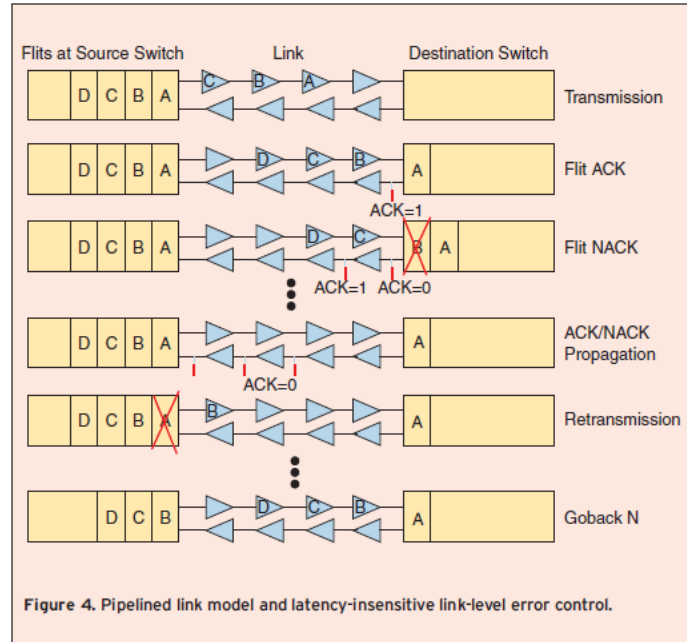


Ex-1005, Fig. 4 (red annotation added).

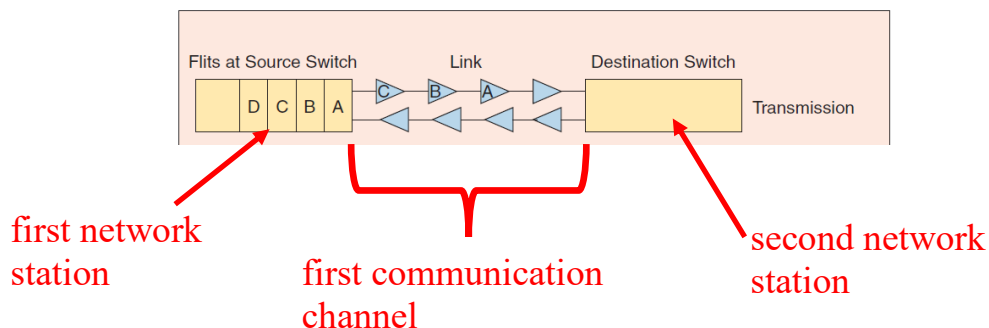
133. [1.e]/[7.e]

Claim language	Evidence and disclosure
[1.e]/[7.e] the data communication network comprising a first network station and a second network station interconnected through a first communication channel,	This element is disclosed by Bertozzi for the same reasons as explained above with regard to element [10.e]. Bertozzi teaches a NoC (“data communication network”) comprising source switch (“first network station”) and a destination switch (“second network station”) interconnected through a first link (communication channel). Ex-1005, Fig. 4, 8 (“Fig. 4 illustrates the link model”, which is equivalent to a pipelined shift register. Pipelining has been used both for data and control lines. The figure also illustrates how pipelined links affect the operation of latency-insensitive link-level error control: the retransmission of a corrupted flit between <i>two</i>

successive switches is represented. **Multiple outstanding flits propagate across the link** during the same clock cycle. When flits are correctly received at the destination switch, an ACK is propagated back to the source, and after N clock cycles (where N is the link length expressed in terms of number of repeater stages) the flit will be discarded from the buffer of the source switch. On the contrary, a corrupted flit is NACKed and will be retransmitted in due time.”); *see also supra*, [10.e].



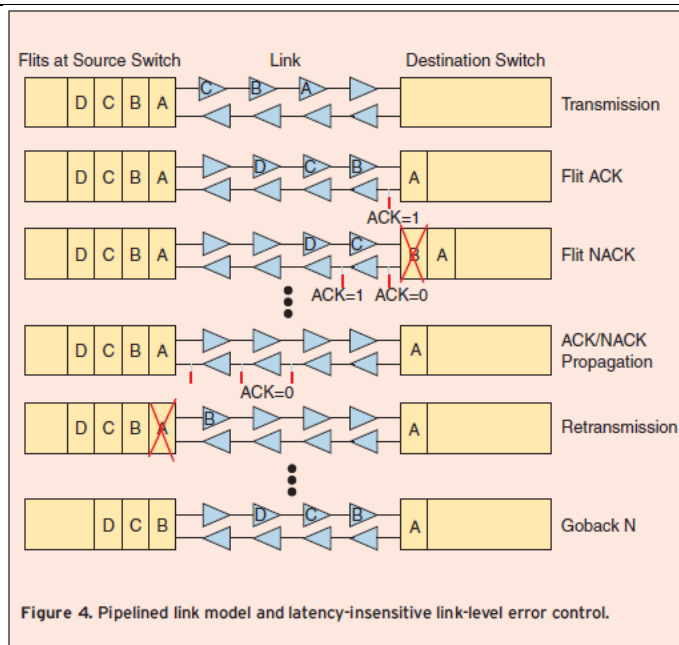
134. I have annotated the first network station, the second network station, and the first communication channel below on Figure 4:



135. [10.f]

Claim language	Evidence and disclosure
[10.f] in response to the identifying act, inserting M*N data storage elements into the data communication network, M being a positive integer, for introducing a delay of M*N cycles on the first communication channel.	This element is disclosed by Bertozzi under both Petitioner Qualcomm’s construction and Patent Owner’s construction. After recognizing that a link has an interconnect-delay problem i.e., identifying links with a delay that exceeds the “pre-defined timing requirements” as discussed above, Bertozzi describes “inserting...data storage elements into the data communication network.” Specifically, Bertozzi discloses inserting “repeaters” into the link to break it up into smaller segments or “pipeline stages.” <i>E.g.</i>, • Ex-1005, 12 (“The complete XpipesCompiler-assisted NoC design flow is depicted in Fig. 9. From the specification of an application, the designer (or a high-level analysis and exploration tool) creates a high-level view of the SoC floorplan, including nodes (with their network interfaces), links and switches. <i>Based on the clock speed target and link routing, the number of pipeline stages for each link is also determined.</i> The information on the network architecture is then specified in an input file for XpipesCompiler”); • Ex-1005, 7 (“A solution to overcome the interconnect-delay problem consists of pipelining interconnects [34, 35]. Wires can be partitioned into segments (or <i>relay stations</i>, which have a function similar to the one of latches in a pipelined data path) whose length satisfies pre-defined timing requirements (e.g., the desired clock speed of the design)...Switch-to-switch links are subdivided into basic segments whose length can be tailored to the desired clock frequency that

	needs to be achieved in the design. In this way, the system operating frequency is not bound by the delay of long links. <i>According to the link length, a certain number of clock cycles is needed by a flit to cross the interconnect.</i>”); • Ex-1005, 8 (“Fig. 4 illustrates the link model, which is equivalent to a pipelined shift register. Pipelining has been used both for data and control lines. The figure also illustrates how pipelined links affect the operation of latency-insensitive link-level error control: the retransmission of a corrupted flit between two successive switches is represented. Multiple outstanding flits propagate across the link during the same clock cycle. When flits are correctly received at the destination switch, <i>an ACK is propagated back to the source, and after N clock cycles (where N is the link length expressed in terms of number of repeater stages)</i> the flit will be discarded from the buffer of the source switch. On the contrary, a corrupted flit is NACKed and will be retransmitted in due time.”) • Ex-1005, 9 (“where N is the link length (expressed as number of basic repeater stages.... Before an ACK is received, <i>the flit has to propagate across the link (N cycles)</i>, an ACK/NACK decision has to be taken at the destination switch (a portion of M cycles), the ACK/NACK signal has to be propagated back (N cycles) and recognized by the source switch (remaining portion of M cycles). During this time, other $2N + M$ flits are transmitted but not yet ACKed.”) • Ex-1005, Fig. 4. An example of such a link is illustrated in Figure 4 of Bertozzi.
--	--



A POSA would understand that the repeaters disclosed by Bertozzi are “data storage elements” under both Petitioner Qualcomm’s proposed construction and Patent Owner’s proposed construction. Bertozzi explains that its repeaters (also called “relay stations”) function similar to a latch in a pipelined data path where each stores a flit (“data element”) for one clock cycle at a time thereby causing a delay of one clock cycle per repeater. Thus, each repeater is “a structure that stores one data element” at a time (Petitioner Qualcomm’s construction) and are an “array of one or more elements that store data” (Patent Owner’s construction). For example:

- Ex-1005, 7 (“Wires can be partitioned into segments (or relay stations, which have a function *similar to the one of latches* in a pipelined data path) whose length satisfies pre-defined timing requirements (e.g., the desired clock speed of the design).”);
- Ex-1005, 7 (“According to the link length, a certain number of clock cycles is needed by a flit to cross the interconnect.”);
- Ex-1005, 8 (“Fig. 4 illustrates the link model, which is *equivalent to a pipelined shift register*.”)

	<i>Pipelining has been used</i> both for data and control lines. The figure also illustrates how pipelined links affect the operation of latency-insensitive link-level error control: the retransmission of a corrupted flit between two successive switches is represented. Multiple outstanding flits propagate across the link during the same clock cycle. When flits are correctly received at the destination switch, an ACK is propagated back to the source, and after N clock cycles (where N is the link length expressed in terms of number of repeater stages) the flit will be discarded from the buffer of the source switch. On the contrary, a corrupted flit is NACKed and will be retransmitted in due time.”); • Ex-1005, 9 (“where N is the link length (expressed as number of basic repeater stages)... the flit has to propagate across the link (N cycles)”).
--	--

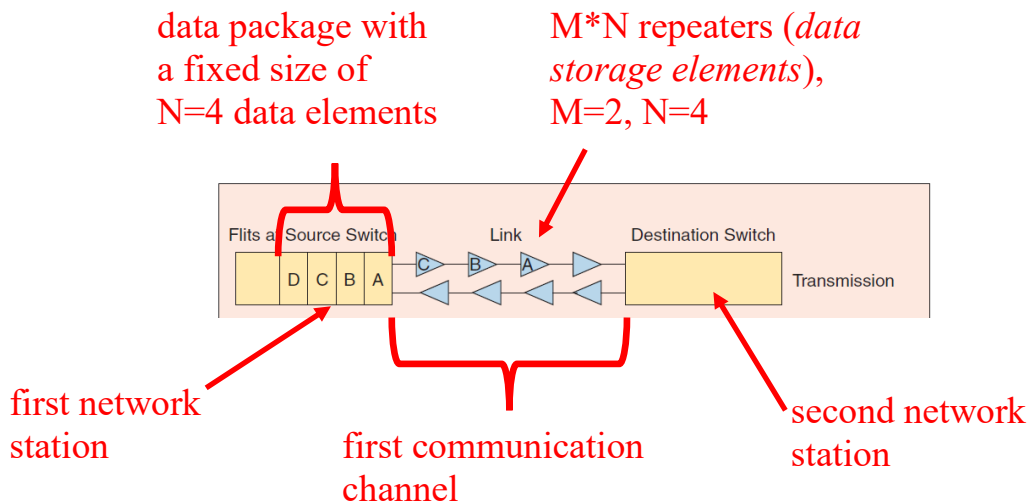
Under Petitioner Qualcomm’s Constructions

136. Bertozzi teaches this element based on Petitioner Qualcomm’s construction of “data storage elements” and “each data package comprising N data elements.”

137. Bertozzi’s Figure 4 discloses inserting “M*N data storage elements ... for introducing a delay of M*N cycles on the first communication channel” as claimed. As I have discussed above, Bertozzi describes data packages with a fixed size of N=4 data elements. *Supra*, [10.pre.c]. In Figure 4, Bertozzi discloses a link with 8 repeaters (“M*N data storage elements” where M=2 and N=4)⁵ have been

⁵ M=2, where 2 is a positive integer.

inserted into the NoC (“data communication network”) between the source switch (“first network station”) and the destination switch (“second network station”). I have annotated this below on Figure 4:



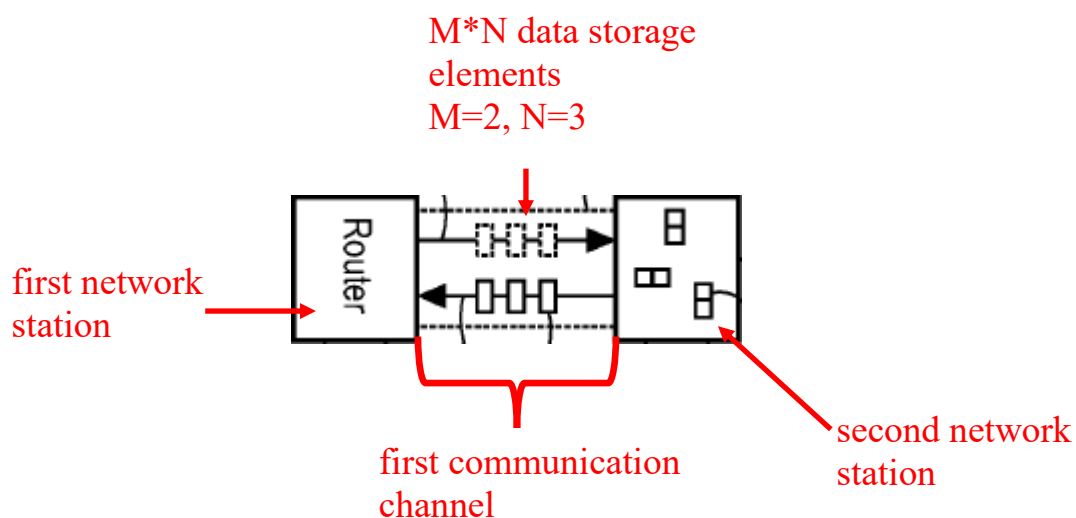
Ex-1005, Fig. 4 (red annotations added).

138. Because Bertozzi explains that each repeater causes a delay of one clock cycle, the repeaters in Figure 4 introduce a delay of 8 data cycles ($M*N$ cycles, where $M=2$ and $N=4$). Ex-1005, 7-8. As I discussed above in the evidence and disclosure table, a POSA would understand that each repeater causes a delay of one clock cycle on the link.

139. Bertozzi’s Figure 4 shows 8 total repeaters ($8=M*N$ where $M=2$ and $N=4$), divided equally between the path from the Source Switch to the Destination Switch and the path from the Destination Switch to the Source Switch. Ex-1005, Fig. 4. This is functionally identical to the embodiment shown in Figure 1 of the

'2893 patent. Figure 1 shows an embodiment where $M=2$, $N=3$ where the data storage elements are split between two subchannels. Ex-1001, 5:63-6:6 (“The network 100 shown in FIG. 1 is arranged to communicate *data packages with size $N=3$* . Other values of N are equally feasible. The first subchannel 152 of the slow communication channel 150 is *extended with $N=3$ data storage elements 160*. Optionally, the second subchannel 154 of *the slow communication channel 150 is also extended with $N=3$ data storage elements 160, in which case $M=2$* . Since the first subchannel 152 and the second subchannel 154 typically will be in each others vicinity, *their respective delay characteristics typically will be similar*, thus requiring the *introduction of data storage elements 160 on both subchannels.*”);

Fig. 1. I have annotated Figure 1 of the '2893 patent in the same manner as Figure 4 of Bertozzi below:



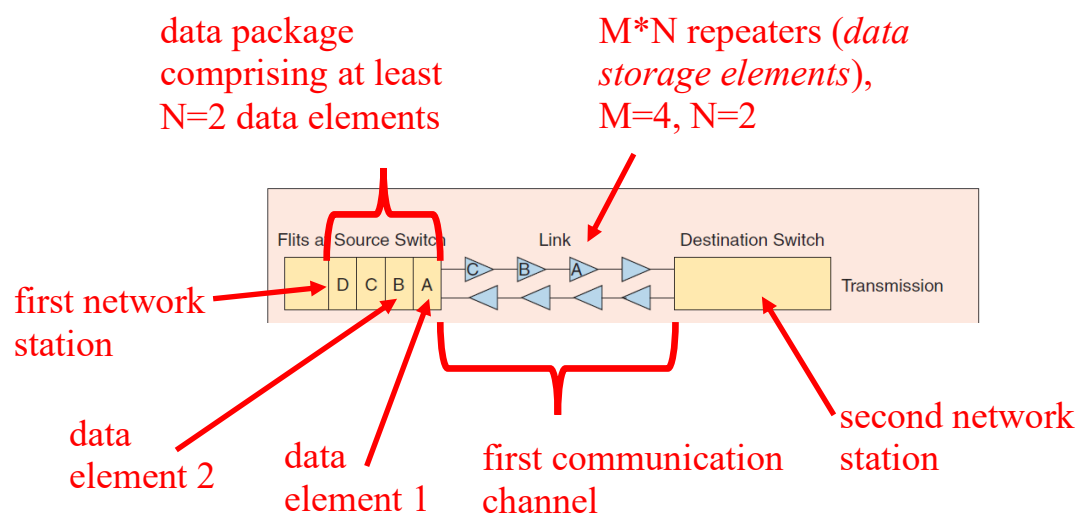
Under Patent Owner's Construction

140. As I discussed above with respect to element [10.pre.c], under Patent Owner's construction, the claim only requires each data package to have "at least N data elements." *Supra*, §VIII; Ex-1027, ¶164. Thus under Patent Owner's construction Bertozzi discloses this element in multiple ways. One, because Patent Owner's construction is broader, Bertozzi discloses this element for the same reasons as under Petitioner's construction. Two, Bertozzi also discloses this element because it teaches packets that include at least two flits, such that each contains at least $N=2$ data elements. *Supra*, [10.pre.c].

141. As an example under Patent Owner's interpretation, Bertozzi's Figure 4 shows a data package that has at least two data elements (e.g., Flit A and Flit B), and it shows that 8 repeaters (" $M*N$ data storage elements" where $M=4$, and $N=2$)⁶ have been inserted into the NoC ("data communication network") between the source switch ("first network station") and the destination switch ("second network

⁶ $M=4$, where 4 is a positive integer. Adding an integer multiple of 2 cycles of delay would thus require an integer multiple of 2 repeaters placed into the communication channel (i.e., 2 repeaters, 4 repeaters, 6 repeaters, 8 repeaters, etc.). Thus, in this instance, $M=4$.

station”).⁷ A POSA would understand that the repeaters (“data storage elements”) are inserted for introducing a delay of (8) $M*N$ data cycles (where $M=4$ and $N=2$) on the link (“first communication channel”). As discussed above, each repeater causes a delay of one cycle on the channel. Ex-1005, 7-8. Therefore, a POSA would recognize that there is a total delay of 8 cycles ($8=M*N$ where $M=4$ and $N=2$) on the link. I have annotated Figure 4 below to show these elements:



⁷ I disagree with Patent Owner’s construction of “at least N data elements” at least because it would allow for this unbounded ability to interpret what a data package is, as illustrated in the example herein where simply setting $N=2$, an integer, would result in such a nonsensical example since the ’2893 specification is clear that data packages have uniform, equal size (fixed size) as I explained previously.

Ex-1005, Fig. 4 (red annotations added). As discussed above in the evidence and disclosure chart, the data storage elements each store a flit (“data element”) for one cycle at a time. Ex-1005, 7-8. A POSA would understand that the repeaters would be an array of one or more elements that stores data.

142. As I have noted above, Bertozzi’s Figure 3 shows a different embodiment with packets having ten flits. Ex-1005, Fig. 3. Under Patent Owner’s broader construction, this claim element [10.f] would have also read on this different embodiment. Under Patent Owner’s construction, Figure 3 shows a data package with (at least) $N=2$ data elements. Under Patent Owner’s construction, the packets would not have to be all of a particular fixed length. In this event, a POSA would at once envisage designing the NoC with the pipeline structure of Bertozzi’s Figure 4, at least because it represents transmission between two consecutive switches with a data package with (at least) $N=2$ data elements. This would result in designing the claimed $M*N$ repeaters and delay cycles. The patent teaches that M can be any positive integer, as I explained above. Therefore, for $M=4$, $N=2$, Figure 4 shows inserting $M*N$ data storage elements (8 elements) into the data communications network, introducing a delay of $M*N$ cycles (8 cycles) on the first communication channel, as claimed.

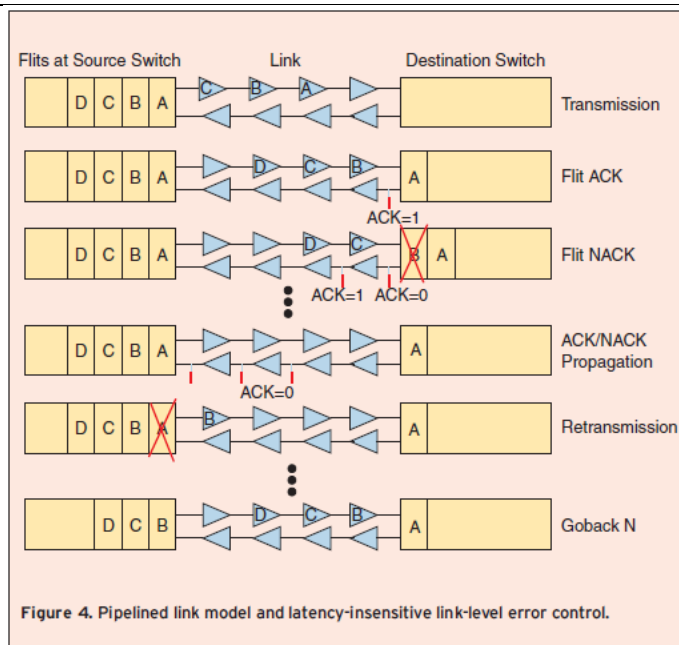
143. [1.f]

Claim language	Evidence and disclosure
----------------	-------------------------

[1.f] the data communication network further comprising M*N data storage elements, M being a positive integer, the data communication introducing a delay of M*N cycles on the first communication channel when the data communication network identifies the first communication channel as having a data transfer delay exceeding a predefined delay threshold	I have been instructed to apply the same scope to element [1.f] as elements [10.e] and [10.f], and I have done so. Therefore, element [1.f] is met by Bertozzi for the same reasons that I discussed above with respect to [10.e] and [10.f].
---	--

144. [7.f]

Claim language	Evidence and disclosure
[7.f] the data communication network further comprising M*N data storage elements, M being a positive integer, for introducing a delay of M*N cycles on the first communication channel,	This element is disclosed by Bertozzi for the same reasons as explained above with regard to element [10.f]. An example of such a link meeting this claim limitation is illustrated in Figure 4 of Bertozzi.



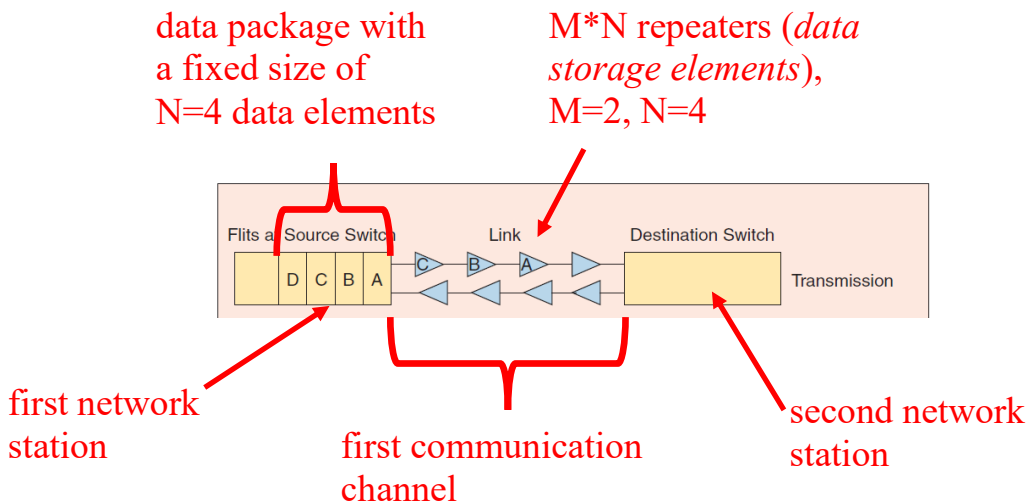
A POSA would understand that the repeaters disclosed by Bertozzi are data storage elements that function similar to a latch in a pipelined data path where each stores a flit (“data element”) for one clock cycle at a time thereby causing a delay of one clock cycle per repeater. Ex-1005, 7 (“Wires can be partitioned into segments (or relay stations, which have a *function similar to the one of latches* in a pipelined data path) whose length satisfies pre-defined timing requirements (e.g., the desired clock speed of the design).”); 7 (“According to the link length, *a certain number of clock cycles is needed by a flit to cross the interconnect.*”); 25 (“Fig. 4 illustrates the link model, which is *equivalent to a pipelined shift register. Pipelining has been used* both for data and control lines. The figure also illustrates how pipelined links affect the operation of latency-insensitive link-level error control: the retransmission of a corrupted flit between two successive switches is represented. Multiple outstanding flits propagate across the link during the same clock cycle. When flits are correctly received at the destination switch, an ACK is propagated back to the source, and after N clock cycles (where N is the link length expressed in terms of number of repeater stages) the flit will be discarded from the buffer of the source switch.

	On the contrary, a corrupted flit is NACKed and will be retransmitted in due time.”); 9 (“where N is the link length (expressed as number of basic repeater stages)... the flit has to propagate across the link (N cycles)”).
--	--

Under Petitioner Qualcomm’s Construction

145. Bertozzi teaches this element based on Petitioner Qualcomm’s construction of “data storage elements” and “each data package comprising N data elements.”

146. A POSA would understand that Bertozzi discloses a NoC (“data communication network”) further comprising $M*N$ repeaters (“data storage elements”) (where $M=2$ and $N=4$)⁸ for introducing a delay of $M*N$ cycles on the link (“first communication channel”) for the same reasons as explained above with regard to element [10.f]. I have annotated this on Figure 4 below:



⁸ $M=4$, where 4 is a positive integer.

Ex-1005, Fig. 4 (red annotations added).

Under Patent Owner's Construction

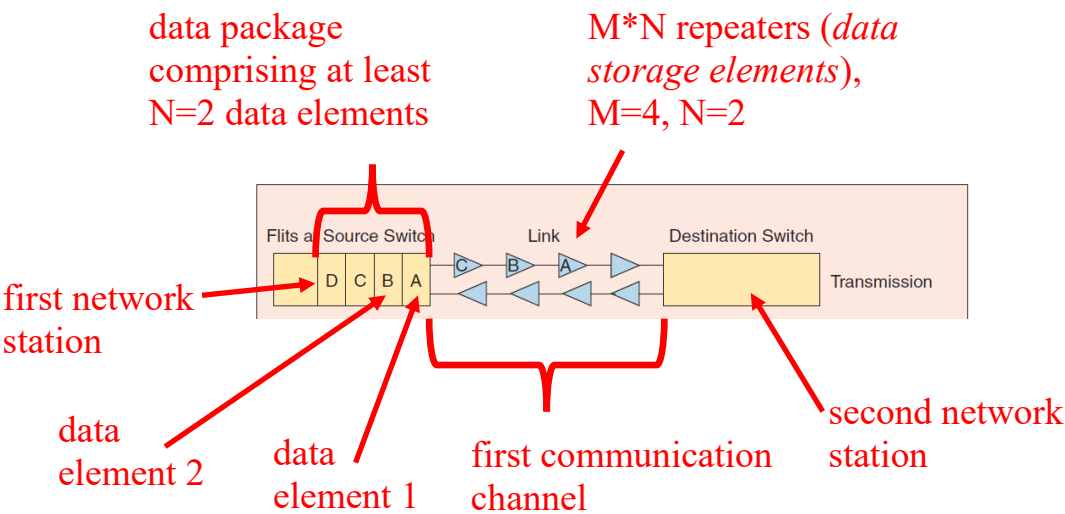
147. Bertozzi teaches this element based on Patent Owner's construction of "data storage elements" and "data package."

148. As I discussed above regarding Petitioner Qualcomm's construction, Bertozzi teaches an embodiment having a fixed size of $N=4$ data elements that would satisfy Patent Owner's construction. This claim element as construed by Patent Owner would be met for the same reasons as discussed above using Petitioner Qualcomm's construction.

149. As I discussed above, Bertozzi also teaches data packages comprising $N=2$ data elements. *Supra*, [1.c]/[7.c]. Bertozzi discloses a NoC ("data communication network") further comprising $M*N$ repeaters ("data storage elements") (where $M=4$, and $N=2$)⁹ for introducing a delay of $M*N$ cycles on the link ("first communication channel") for the same reasons as I explained above

⁹ $M=4$, where 4 is a positive integer. Adding an integer multiple of 2 cycles of delay would thus require an integer multiple of 2 repeaters placed into the communication channel (i.e., 2 repeaters, 4 repeaters, 6 repeaters, 8 repeaters, etc.). Thus, in this instance, $M=4$.

with regard to element [10.f].¹⁰ I have annotated Figure 4 below to show these elements:



Ex-1005, Fig. 4 (red annotations added).

150. [7.g]

Claim language	Evidence and disclosure
[7.g] wherein each communication channel of the plurality of communication	Bertozzi discloses this limitation. A POSA would understand that each of the plurality of link (“communication channels”) disclosed by Bertozzi are bidirectional comprised of two subchannels, one for

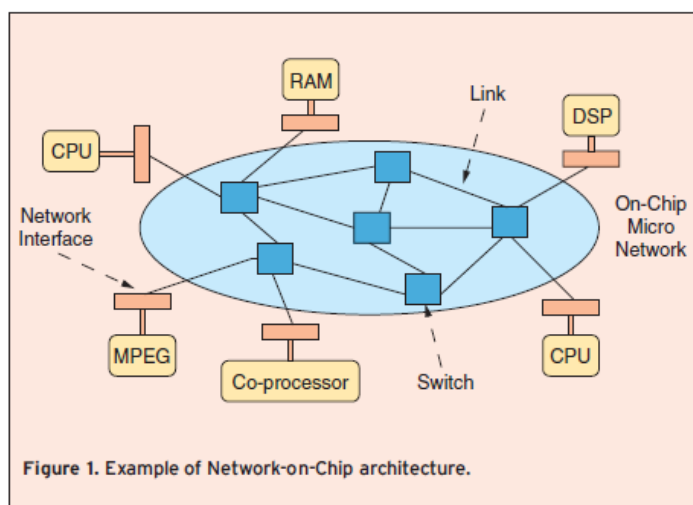
¹⁰ I disagree with Patent Owner’s construction of “at least N data elements” at least because it would allow for this unbounded ability to interpret what a data package is, as illustrated in the example herein where simply setting N=2, an integer, would result in such a nonsensical example since the ’2893 specification is clear that data packages have uniform, equal size (fixed size) as I explained previously.

channels comprises: a first subchannel between an input port of a first associated network station and an output port of a second associated network station; and a second subchannel between an input port of the second associated network station and an output port of the first associated network station, and

each direction of communication between the network stations. As shown in Bertozzi Figure 4, such communication would use two subchannels: a first link (“subchannel”) between the input port of the first network station to the output port of the second network station and second link (subchannel) between the input port of the second network station to the output port of the first network station.

A POSA would have understood that the subchannels described above exist in all the links (and embodiments of) Bertozzi’s network. In the event that Patent Owner argues that each channel does not include subchannels, a POSA would have at once envisaged that the subchannel taught by Bertozzi (e.g., Figure 4) can be used in all links of (and embodiments of) Bertozzi’s network.

For example, a POSA would understand that each of the cores (functional blocks) shown in Figure 1 provide the ability to both send and receive messages in order to implement the communication protocol between network stations discussed in Bertozzi and further illustrated below. Thus, their network interfaces include both input and output ports, along with associated subchannels.

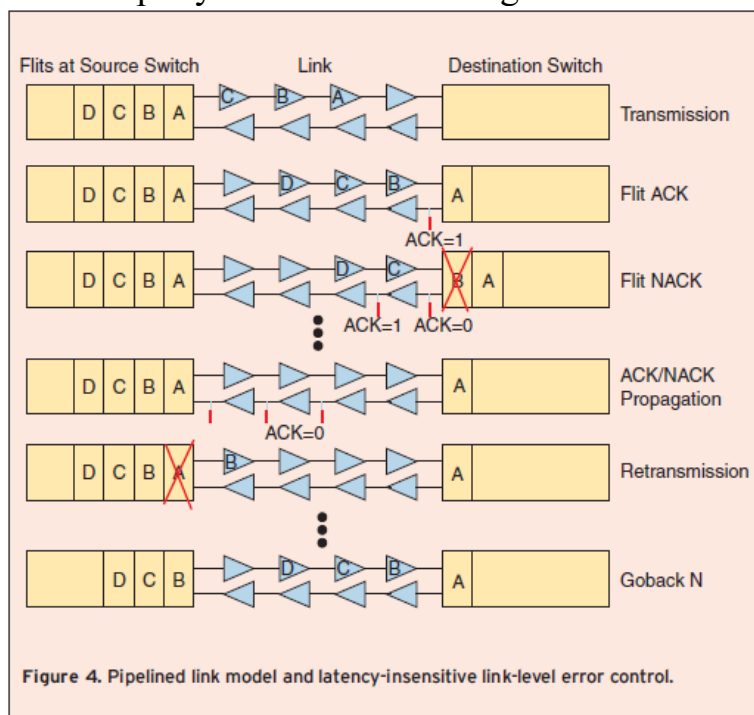


Ex-1005, Fig. 1. A CPU (Central Processing Unit), RAM (Random Access Memory), DSP (Digital Signal Processor), Co-Processor, and MPEG (Moving Picture

Experts Group, this would typically be a display) all require the ability to send and receive information. For example, the CPU may send a read request to the RAM. The RAM in turn would receive the read request and respond by sending requested data to the CPU.

A POSA would thus understand that the links such as those in Figure 1 need to be bidirectional, meaning that that there would need to be a first link (“subchannel”) between the input port of a first associated network station and an output port of a second associated network station and a second link (“subchannel”) between an input port of the second associated network station and an output port of the first associated network station.

An exemplary link is shown in Figure 4 of Bertozzi.

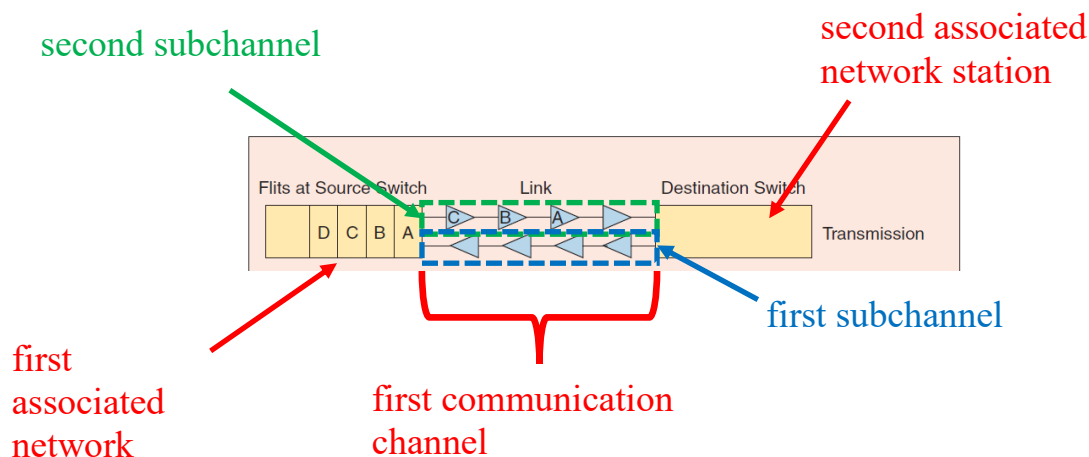


A POSA would understand that a link taught by Bertozzi is connected between the output port of one network station and the input port of another network station as the switches and network interfaces taught by Bertozzi have input and output ports. Ex-1005, Figs. 5, 7.

	Bertozzi also teaches that there are acknowledgements sent over the links between network stations. Bertozzi uses “distributed error detection with link level retransmission” for error recovery. Ex-1005, 6. A POSA would understand that under this approach, for each flit received by a second associated network station from a first associated network station over a link, the second associated network station must send an acknowledgement message (e.g., ACK or NACK) to the first associated network station over the link. For example: • Ex-1005, 7 (“The retransmission policy implemented in Xpipes is GO-BACK-N. Flits are transmitted continuously and the transmitter does not wait for an ACK after sending a flit. Such an ACK is received after a round-trip delay. When a NACK is received, the transmitter backs up to the flit that is negatively acknowledged and re-sends it in addition to the N succeeding flits that were transmitted during the round-trip delay. At the receiver, the N-1 received flits following the corrupted one are discarded regardless of whether they were received error-free or not. GO-BACKN trades-off inefficiency in bus usage (retransmission of many error-free flits) with a moderate implementation cost, and was preferred to a selective-repeat scheme, <i>wherein only those flits that are negatively acknowledged</i> are retransmitted but more buffering resources are required at the destination switch.”); • Ex-1005, 8 (“Fig. 4 illustrates the link model, which <i>is equivalent to a pipelined shift register</i>. Pipelining has been used both for data and control lines. The figure also illustrates how pipelined links affect the operation of latency-insensitive link-level error control: the retransmission of a corrupted flit between two successive switches is represented. Multiple outstanding flits propagate across the link during the same clock cycle. <i>When flits are correctly</i>
--	---

	<i>received at the destination switch, an ACK is propagated back to the source, and after N clock cycles (where N is the link length expressed in terms of number of repeater stages) the flit will be discarded from the buffer of the source switch. On the contrary, a corrupted flit is NACKed and will be retransmitted in due time.”);</i> • Ex-1005, Fig. 4.
--	--

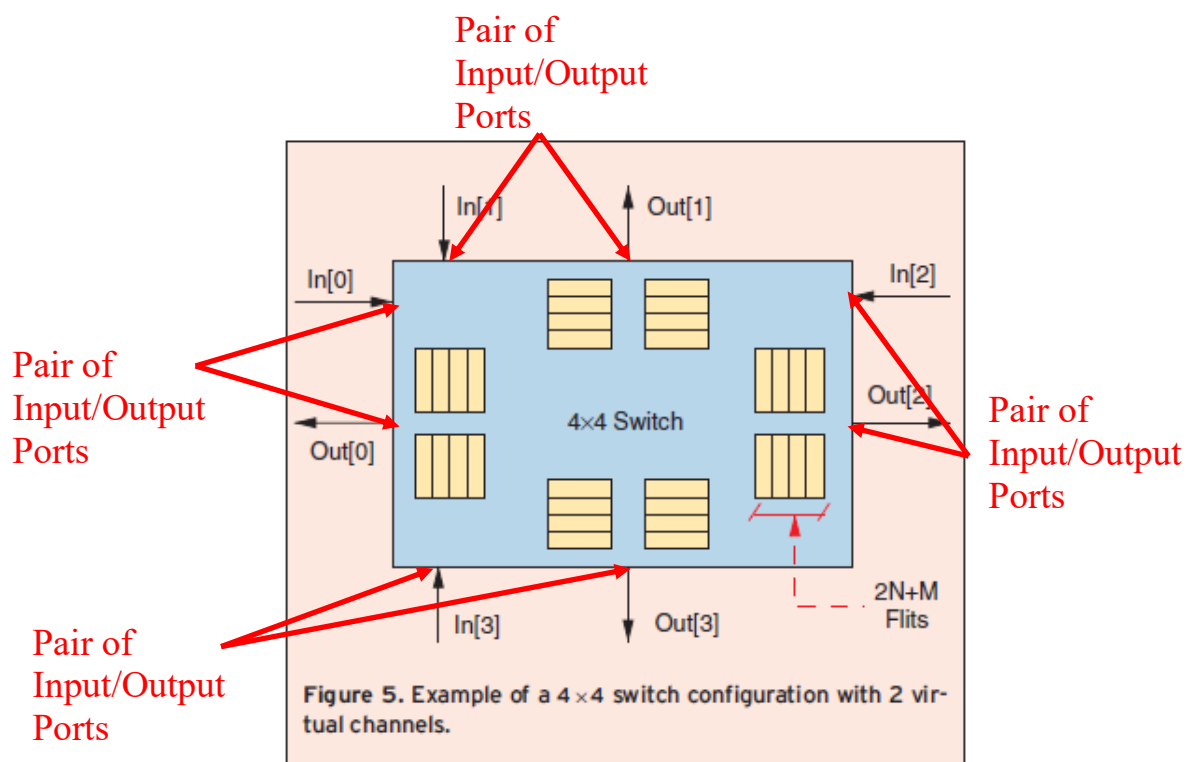
151. I have annotated an example link from Figure 4 below, which depicts a first link (subchannel) directed from the output of the destination switch (second associated network station) to the input of the source switch (first associated network station) and a second link (subchannel) directed from the output of the source switch (first associated network station) to the input of the destination switch (second associated network station):



Ex-1005, Fig. 4 (annotations added).

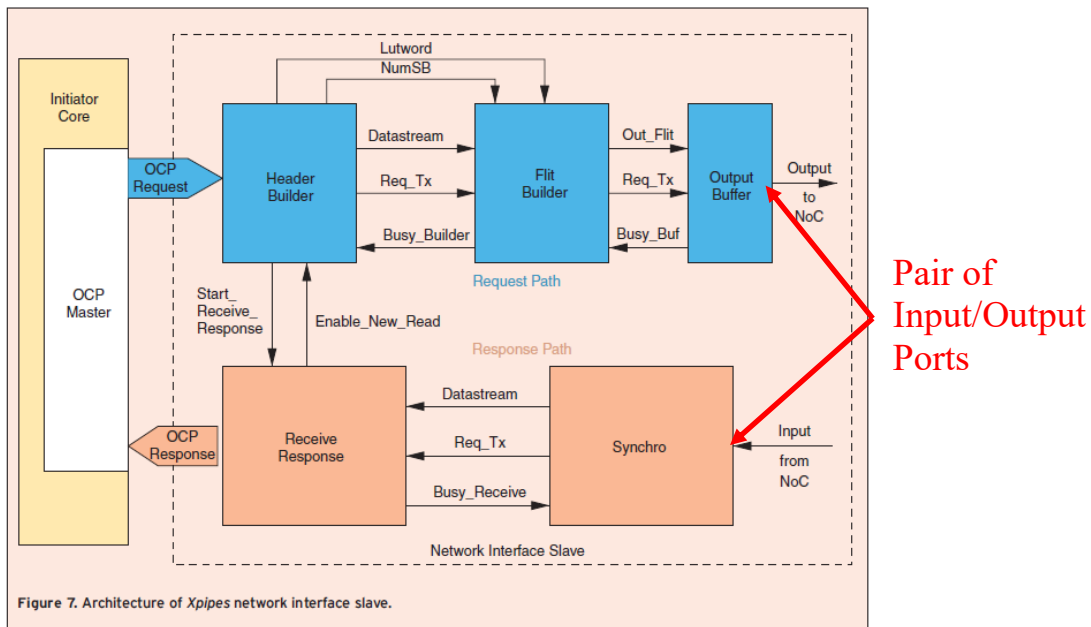
152. Bertozzi describes that these subchannels connect to *input ports* and *output ports* in the network interfaces and switches (*network stations*). Ex-1005, Figs. 5, 7. For example, Bertozzi's Figure 5 (switch) and Figure 7 (network interface) show that the network stations each have at least one pair of input/output ports. A POSA would have used the same link to connect the paired input and output ports of Bertozzi's network stations. Thus, A POSA would understand from Figures 5 and 7 that the paired input and output ports were intended to be connected via the same link to the same network station.

153. I have annotated the input and output ports of the switch shown in Figure 5 below:



Ex-1005, Fig. 5 (red annotations added).

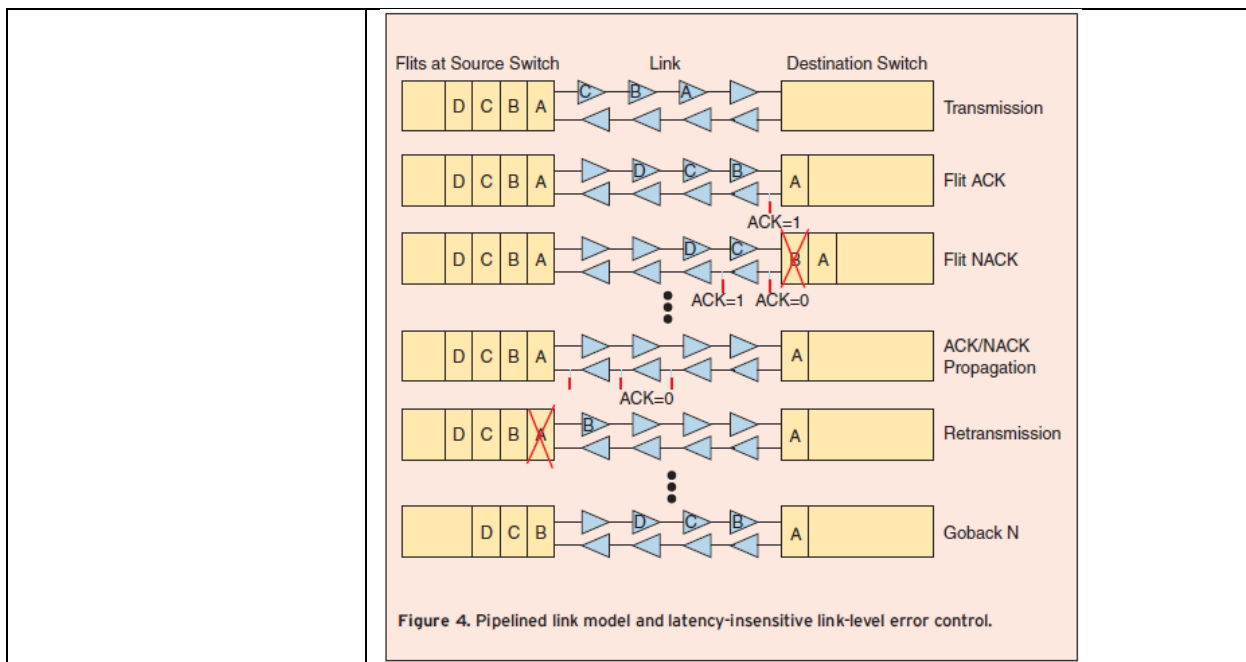
154. I have also annotated the input and output ports of the network interface shown in Figure 7 below.



Ex-1005, Fig. 7 (red annotations added).

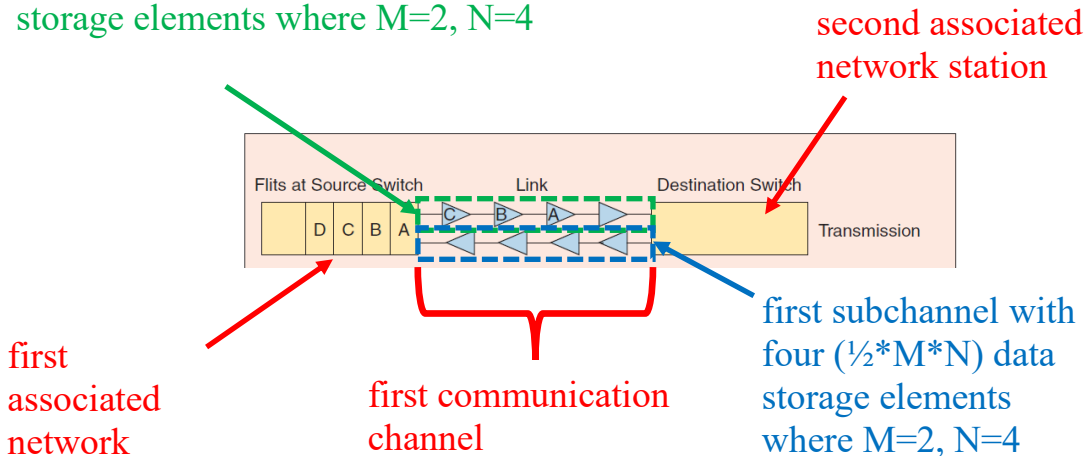
155. [7.h]

Claim language	Evidence and disclosure
[7.h] wherein the first subchannel and the second subchannel of the first communication channel each comprise a delay introduced by $\frac{1}{2} * M * N$ data storage elements	Bertozzi discloses this element under both Petitioner Qualcomm's constructions and Patent Owner's constructions. At Figure 4, Bertozzi shows the first subchannel and the second subchannel of the first link ("communication channel") comprise a delay introduced by $\frac{1}{2} * M * N$ of the repeaters ("data storage elements"). Ex-1005, Fig. 4.



156. Under both Petitioner Qualcomm’s and Patent Owner’s constructions, this element taught by the embodiment of Bertozzi discussed above in [1.c]/[7.c] where $M=2$ and $N=4$, both the first subchannel and the second subchannel of the first link (“communication channel”) comprise a delay introduced by four ($\frac{1}{2} * M * N$) repeaters (“data storage elements”). Ex-1005, Fig. 4. I have annotated Figure 4 below to show this:

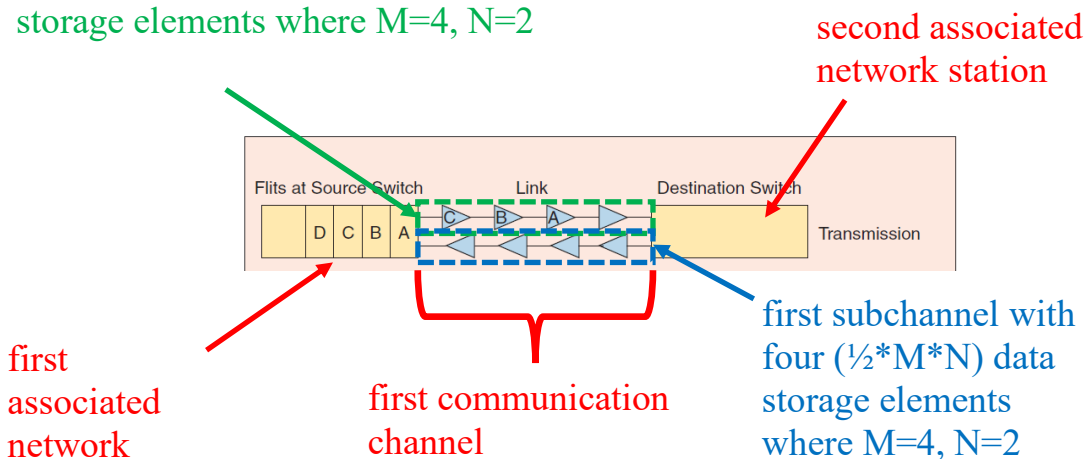
second subchannel with four ($\frac{1}{2} * M * N$) data storage elements where $M=2$, $N=4$



Ex-1005, Fig. 4 (annotations added).

157. Bertozzi also teaches this element under Patent Owner's construction, where $M=4$ and $N=2$, and both the first subchannel and the second subchannel of the first link ("communication channel") comprise a delay introduced by 4 ($\frac{1}{2} * M * N$) repeaters ("data storage elements"). Ex-1005, Fig. 4. I have annotated Figure 4 below to show this:

second subchannel with four ($\frac{1}{2} * M * N$) data storage elements where $M=4$, $N=2$



Ex-1005, Fig. 4 (annotations added).

158. Dependent Claim 2:

Claim language	Evidence and disclosure
[2]: The integrated circuit as claimed in claim 1, wherein each communication channel of the plurality of communication channels comprises: a first subchannel between an input port of a first associated network station and an output port of a second associated network station; and a second subchannel between an input port of the second associated network station and an output port of the first associated network station for enabling duplex communication between the two associated network stations.	A POSA would understand that bidirectional links enable duplex communication between two associated network stations. Therefore, Bertozzi teaches dependent claim 2 for the reasons explained above with respect to elements [7.g].

159. Dependent Claim 3:

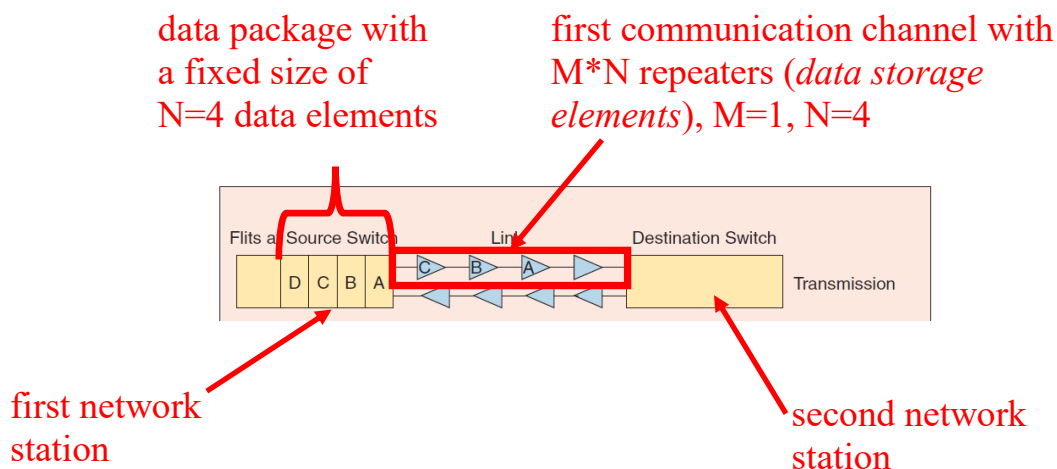
Claim language	Evidence and disclosure
[3] The integrated circuit as claimed in claim 2, wherein the M*N data storage elements are distributed over the first subchannel and the second subchannel of the first communication channel.	Bertozzi teaches dependent claim 3 for the reasons explained above with respect to element [7.h].

160. Dependent Claim 6:

Claim language	Evidence and disclosure
[6] The integrated circuit as claimed in claim 1, wherein M=1.	Bertozzi teaches this element for the same evidence as discussed in elements [10.f] and [1.f].

161. Bertozzi teaches this element. Claim 6 is met for the same reasons as discussed above regarding element [10.f] and [1.f], except that a POSA would understand that the “first communication channel” to be the link identified below

in Bertozzi Fig. 4:

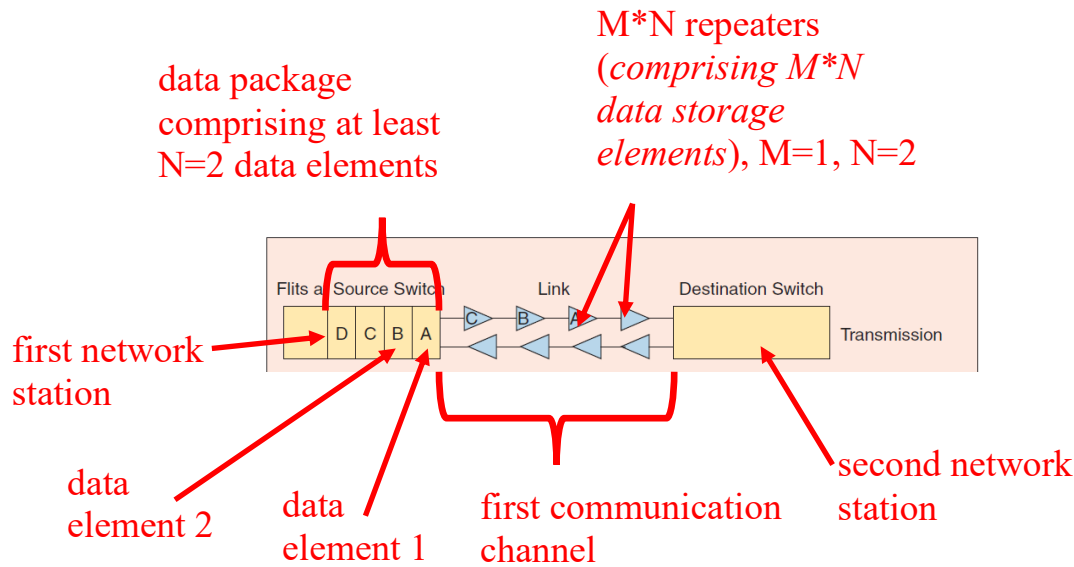


Ex-1005, Fig. 4 (excerpted, annotated). A POSA would understand such a link to be a communication channel.

162. The '2893 patent indicates that an example of a communication channel is "wires." Ex-1001, 1:24-35 ("communication channels, *e.g. wires*"). The link identified above ("first communication channel") is a segmented wire. Ex-1005, 7 ("A solution to overcome the interconnect-delay problem consists of pipelining interconnects [34, 35]. *Wires can be partitioned into segments* (or relay stations, which have a function similar to the one of latches in a pipelined data path) whose length satisfies pre-defined timing requirements (e.g., the desired clock speed of the design).") The '2893 patent does not require a "communication channel" to be bidirectional or duplex. Ex-1001, 5:1-5 ("The data network 100 shown in FIG. 1 is a duplex network, i.e. it allows simultaneous data communication in two directions. *It is emphasized, however, that the present*

application is equally applicable to simplex networks, i.e., networks that do not support simultaneous two-way communication.”).

163. Alternatively, as discussed above with respect to element [10.pre.c], under Patent Owner’s interpretation that the claim only requires each data package to have “at least N data elements.” Ex-1026, ¶164. Bertozzi discloses packets that include at least two flits, such that each contains at least $N=2$ data elements. See my discussion of element [10.pre.c] above. Similarly, claim 1 for which claim 6 depends requires “comprising $M*N$ data storage elements...the data communication introducing a delay of $M*N$ cycles on the first communication channel.” Ex-1001, 10:43-46. As one example under Patent Owner’s interpretation illustrated below, Bertozzi’s Figure 4 discloses 2 repeaters (*$M*N$ data storage elements*, where $M=1$, and $N=2$) that introduce a delay of 2 data cycles (*$M*N$ cycles*, where $M=1$, and $N=2$). Ex-1005, Fig. 4 and 7-8. Thus, under Patent Owner’s construction, Bertozzi discloses the integrated circuit of claim 1 with $M=1$:



Ex-1005, Fig. 4 (excerpted, annotated).

XII. GROUND 2: BERTOZZI ALONE OR IN VIEW OF JANTSCH OR DUATO RENDERS CLAIM 1-3, 6-7, AND 10 OBVIOUS

164. It is my opinion that Bertozzi teaches all of the elements of claims 1-3, 6-7, and 10 for the reasons I explained above with regard to Ground 1. However, it is also my opinion that even if it did not, a POSA would have understood what is claimed to be a straightforward, predictable variation of Bertozzi. A POSA would have been motivated, with a reasonable expectation of success, to make this variation in view of the teachings of Bertozzi alone, or in view of Jantsch or Duato.

[10.pre.c], [1.c] and [7.c] each data package comprising N data elements including a data element comprising routing information for the network stations, N being an integer of at least two,

165. Even if Patent Owner were to argue that Bertozzi discloses designing an integrated circuit for communicating data packages of different lengths (or

different lengths than the four data elements shown in Figure 4), as explained above Bertozzi still anticipates Claim 10 under PO's construction. This is because under PO's construction, the total number of data elements in each packet (data package) does not need to be the "N" recited in the claim—each data packet just needs to have *at least a* "N" of two data elements somewhere in the packet. *See supra*, §VIII, [10.pre.c].

166. However, designing an integrated circuit for communicating fixed length data packages (as Petitioner Qualcomm's construction provides), such as the N=4 data elements shown in Bertozzi, was also obvious in view of the teachings of Bertozzi and Duato or Jantsch. Each of Jantsch and Duato teaches designing and using systems for fixed packet size.¹¹ Ex-1006, 56 (NoC systems for a "fixed

¹¹ As I noted above regarding Fixed-Length Packets, a POSA would understand fixed-length packets mean that each data packet is the same size or length. *See* Ex-1015, 3 ("ATM packets are fixed-sized packets with data fields of 48 bytes. Five bytes are used for header, so the length of the whole packet is 53 bytes."); *see also* Ex-1035, 4 ("fixed length: Pertaining to a record of field that has a constant length, regardless of the specific data contained in it. Filler characters may be used to maintain the fixed length. *Contrast*: variable length. *See also*: fixed format.") and 3

packet size”); Ex-1009, 32 (router networks for “fixed-length packets”). A POSA would have found it obvious to design, and for the integrated circuit disclosed by Bertozzi to use, data packages, each having a fixed-length of four data elements like the data package shown in Figure 4 (identified A-D). Ex-1005, 6-8, Fig. 4.

167. In fact, Bertozzi itself discloses the use of a “fixed-length” packet. Ex-1005, 5 (“Messages that have to be transmitted across the network are usually partitioned into fixed-length packets.”). And with good reason, as it was known and disclosed in Jantsch, there were two ways to communicate—fixed and variable-length packets—and a fixed-length packet provided efficiencies in comparison to variable-length packets. Ex-1006, 56 (“a fixed packet size with a payload of eight will be efficient compared to a variable packet size.”); Ex-1009, 32 (“each message may be partitioned into fixed-length packets.”).

168. For example, it was known that fixed-length packets provided the benefit of keeping packets moving within a priority based NoC such as Bertozzi’s, which uses arbitrators within its switches. Ex-1005, 10 (“Arbitration is carried out when the tail flit of the preceding packet is received, so that all other flits of a packet can be propagated without contention related delay at this state.”); Fig. 6. In

(fixed format: A file organization in which all logical records in the file of fixed length. *Synonym:* F format. *Contrast:* variable format.”) (emphasis in original).

contrast, variable-length packages, which were known to include a few long packages, were known to occupy a router or switch for an extended period of time. Ex-1009, 98-99 (“When traffic consists of messages of *very different lengths*, a *few long messages may reserve all of the virtual channels of a given physical channel*. If a short message requests that channel, it will have to wait for long. *A simple solution consists of splitting long messages into fixed length packets*. This approach reduces waiting time for short messages but introduces some overhead because each packet carries routing information and has to be routed.”).

169. In addition, fixed length packets were known to improve deadlock recovery. Ex-1009, 67 (“However, when long packets become deadlocked recovering from deadlock may take a while. As a consequence, other deadlocked packets have to wait a considerable amount of time before the central buffer becomes available, increasing latency and reducing throughput. In these situations, latency becomes much less predictable. Additionally, the optimal value of the timeout threshold for deadlock detection heavily depends on packet length unless some type of physical channel monitoring of neighboring nodes is implemented. *Therefore, short messages or messages split into fixed-length shorter packets may be preferred.*”).

170. Fixed length packets also would have been expected to make it easier to allocate buffer space in network stations. Ex-1009, 35 (“In circuit switching, the

complete message is transmitted after the circuit has been set up. Alternatively, the message can be partitioned *and transmitted as fixed-length packets*, e.g., 128 bytes. The first few bytes of a packet contain routing and control information and are referred to as the packet header. Each packet is individually routed from source to destination. *A packet is completely buffered* at each intermediate node before it is forwarded to the next node. This is the reason why this switching technique is also referred to as store-and-forward (SAF) switching. The header information is extracted by the intermediate router and used to determine the output link over which the header is to be forwarded.”).

171. Fixed length packets would also have predictably allowed simpler hardware implementation of various NoC components. In such cases, network stations do not need to deal with variations in packet length that can lead to more complex designs. An example of this is error checking. Because each packet is the same size, meaning that it had the same number of flits, the network stations know to expect data formatted a certain way at a certain time. For example, if the header bit is not set when expected, the network station knows an error has occurred and can ask for retransmission.

172. A POSA would have been motivated, with a reasonable expectation of success, to use fixed-length packets in Bertozzi’s NoC, for at least four reasons.

173. **One**, Bertozzi explicitly teaches that “messages ... *are usually partitioned into fixed-length packets.*” Ex-1005, 5. This would indicate to a POSA that—to the extent Bertozzi were found to not expressly teach designing a system for fixed-length packets—it was routine and well within the ordinary skill of a POSA at the time to design a system for fixed-length packets. Ex-1005, 5; *see also e.g.*, 56 (“a *fixed packet size* with a payload of eight will be efficient compared to a variable packet size.”); Ex-1009, 32 (“each message may be partitioned into *fixed-length packets.*”).

174. **Two**, the technology taught by Jantsch and Duato is closely related to the technology in Bertozzi. For example, Bertozzi teaches a packet switched NoC interconnect for SoCs (Ex-1005, 18), and Chapter 5 of Jantsch, the referenced portion, is titled “On Packet Switched Networks for On-Chip Communication.” Ex-1006, 47. Duato is a textbook teaching interconnection networks, which includes “very large-scale integrated (VLSI) circuits.” Ex-1009, 3. SoCs are a type of VLSI circuit. Ex-1001, 1:8-11.

175. **Three**, as discussed above, a POSA would have known that a network using fixed-length packets can be implemented using simpler hardware. This is because network stations in a fixed-length packet network do not need to be designed to deal with variations in packet length, which can add complexity as network stations need to be able to operate without knowing when a packet is

going to end. Fixed-length packets were known to be predictable, such that network stations could be designed to expect data formatted in a certain way at a certain time. This was known to simplify various processes, such as error checking. Further, a POSA would understand that hardware designed to operate using variable-length packets could easily transport fixed-length packets with little to no modification.

176. **Four**, using fixed length packets within a NoC like Bertozzi's would have been routine and well within the knowledge of an ordinary artisan in the field. Fixed-length packets were a well-known option in the art for a POSA designing a packet-switching network. *E.g.*, Ex-1005, 5 ("Message that have to be transmitted across the network are usually partitioned into ***fixed-length packets***"); Ex-1006, 56 ("choosing a ***fixed packet size*** with a pay load of eight will be efficient as compared to a variable packet size."); Ex-1009, 32 ("each message may be partitioned into fixed-length packets"); Ex-1015, 1 ("This paper examines the technical merits of ***fixed-sized packets*** versus variable-sized packets....").

[10.pre.d], [1.d], and [7.d] the plurality of network stations comprising a plurality of data routers and a plurality of network interfaces, each of the data routers being coupled to a functional block via a network interface

177. As discussed above in ground 1, Bertozzi discloses network stations comprising "data routers." *Supra*, [10.pre.d]. To the extent that Patent Owner mischaracterizes Bertozzi's network and argues it discloses only switches and not

routers, it would have been obvious to use data routers in Bertozzi's network as recited by claim elements [10.pre.d], [1.d], and [7.d]. Bertozzi uses the terms routers and switches interchangeably: "the basic idea [of the disclosed NoC] ... envisions on-chip ***router (or switch)-based networks*** on which packetized communication takes place, as depicted in Fig. 1." Ex-1005, 2. Even if they are distinct (and they are not in the context of Bertozzi), a POSA would have been motivated to use data routes instead of switches given Bertozzi's express disclosure and would have known it would have known it would be a simple substitution of one known element for another to obtain a predictable result.

178. A POSA would have had a reasonable expectation of success.

Bertozzi itself teaches that both routers and switches may be used in NoCs. Ex-1005, 2 ("The basic idea is borrowed from traditional large-scale multi-processors and wide-area networks domain, and envision ***on-chip router (or switch)-based networks*** on which packetized communication takes place, as depicted in Fig. 1.>").

A POSA would expect a router to work in Bertozzi. First, it would have been a simple substitution of one known element for another that performs the same function of transporting packets along the network. *Id.* Second, routers were known to be able to perform the same "source routing" technique as disclosed by Bertozzi. *See* Ex-1005, 5 ("The header contains the necessary routing and network control information (e.g., the source and destination address). When ***source***

routing is used, the destination address is ignored and replaced with a route field that specifies the route to destination. This overhead in terms of packet header is counterbalanced by a simpler routing logic at the network switches: they simply have to look at the routing fields in the packets and route them over the specified switch output port.”). For example, the Aethereal NoC uses routers. *E.g.*, Ex-1007, (“The network on chip as is exemplified by Figure 1, consists of two components: the *routers* and the network interfaces (NI).”). It also uses source routing. Ex-1007, 2 (“We select[ed] *source routing* as an addressing scheme....In source routing the path on which the packet travels is included in the header of a packet. In Aethereal, this path is a list of destination ports, from which each router on the path removes the first element for its own use.”). Thus, a POSA would have a reasonable expectation of success.

[1.f] the data communication network further comprising M*N data storage elements, M being a positive integer, the data communication introducing a delay of M*N cycles on the first communication channel when the data communication network identifies the first communication channel as having a data transfer delay exceeding a predefined delay threshold.

179. As discussed above, Bertozzi discloses “the data communication network identifies the first communication channel as having a data transfer delay exceeding a predefined delay threshold” as this element has been applied by Patent Owner. *See supra*, Ground 1 [1.f]. To the extent this element requires that the data communication network itself identify a delay during the operation of the claimed

integrated circuit (rather than a delay being identified during the design stage)—which I am told is contrary to Patent Owner’s application in its litigation against Petitioner Samsung—the detection of such delays in operational networks would have been obvious to a POSA. This is at least because Bertozzi recognizes the variability of the wire delays attributed to deep submicron effects and signal integrity issues, and contemplates solutions including self-synchronization techniques and on-chip networks that allow electrical parameters of the wires to be optimized and controlled through performance optimization techniques. For example, Bertozzi discloses:

While gate delays scale down with technology, *global wire delays typically increase or remain constant as repeaters are inserted*. It is estimated that in 50 nm technology, at a clock frequency of 10 GHz, *a global wire delay can be up to 6–10 clock cycles* [3]. Therefore, limiting the on-chip distance travelled by critical signals will be key to guarantee the performance of the overall system, and will be a common design guideline for all kinds of system interconnects. Synchronization of cores on future SoCs will be unfeasible *due to deep submicron effects (clock skew, power associated with clock distribution trees, etc.)*, and an alternative scenario consists of *self-synchronous cores that communicate with one another through a network-centric architecture* [14]. Finally, *signal integrity issues* (cross-talk, power supply noise, soft errors, etc.) will lead to more transient and permanent failures of signals, logic values, devices *and interconnects*, thus raising the reliability concern for on-chip communication [15]. In many cases,

on-chip networks can be designed as regular structures, allowing electrical parameters of wires to be optimised [sic] and well controlled. This leads to *lower communication failure probabilities*, thus enabling the use of low swing signalling [sic] techniques [16], and to the capability of exploiting *performance optimisation [sic] techniques* such as wavefront pipelining [17]

Ex-1005, 2-3.

Finally, the design of communication (switch-to-switch and network interface-to-switch) *links represents a critical issue with respect to the system performance.* As geometries shrink, gate delay improves much faster than the delay in long wires. Therefore, the *long wires increasingly determine the maximum clock rate*, and hence performance, of the entire design. The problem becomes particularly serious for domain-specific heterogeneous SoCs, where the *wire structure is highly irregular and may include both short and extremely long switch-to-switch links. Techniques have to be devised to decouple link throughput from its length and to allow functionally correct operation of the switches in presence of misaligned inputs due to the unequal length of the input links.*”

Ex-1005, 6.

180. For example, it was well-known that synchronization of certain networks and types of communications require packets to arrive at network stations “at precisely scheduled time slots.” Ex-1014, 11:23-27 (“All the operations concerning receiving routing enqueueing/dequeueing take place concurrently at a rate that matches the total packet throughput rate. This performance can be achieved

because all packets are of equal size and *arrive at precisely scheduled time slots.*")

This is just what is described in the '2893 patent. See Ex-1001, 1:47–61

(describing prior art networks in which data packets are received “during a predefined clock cycle”).

181. It was also well-known to initialize operational networks to determine round-trip delays *between modules*. Ex-1014, 2:43-48 (“initializing a network to determine round-trip delays between adjacent nodes as well as breaking cycles to avoid deadlock situations”) and 2:53-56 (“provide a temporal alignment buffer to provide an adjustment in the round-trip delay for packets between given nodes.”). As such, in networks with precisely scheduled time slots, it was known to periodically detect delay in a transmission channel (e.g., using “probe” signals) (“the receiving of either signal, "probe" or "probe-echo", *provides the ability to set the temporal alignment delay so that the probe words arrive synchronized to the respective routing cycles.*” Ex-1014, 16:11-15) that would result in a packet being received in an inappropriate time slot (“*a data transfer delay exceeding a predefined delay threshold*”) and to use data storage elements like those described by Bertozzi to adjust the delay to be an “integral multiple of the packet transmission time” (“*a delay of $M*N$ cycles*”) so that the packet arrives in an appropriate time slot. See:

- Ex-1014, 11:47–67 (“A channel module on the receiving end of a packet transmission receives the 96 bits of the packet data with appended postfix data as a serial bit stream. The channel maintenance subsystem at decoding logic 68 of the receiving channel modules decodes the incoming packet. The encoding logic 58 is implemented at design time to follow a predetermined encoding algorithm. The decoding logic of the receiving channel module must be selected for decoding the chosen encoding scheme. The packet is then aligned to 16-bit double words at packet alignment circuit 69. The packet alignment circuit 69 includes a queue to absorb clock jitters and the *variable delay element, the temporal alignment buffer 69, that is used to adjust the round-trip delay to an integral multiple of the routing cycle.* The temporal alignment buffer 69 is used to *adjust the signal transit time to be in integral multiple of the packet transmission time.* This is achieved by *adding the delay element in the receive data path*, which is essentially *a 16-bit wide shift register of depth 0-5.* The depth is set during the initialization procedure *when the round-trip delay is calculated and the packet transmission time is known.”*),
- Ex-1014, 12:38–44 (“FIG. 12 is provided to illustrate a graphical demonstration of the conveyor belt nature of the slotted packet exchange between linked channel modules. As was described, *the transmission time between adjacent channel modules is adjusted by means of the temporal alignment buffer* to provide exactly an integral number of packet transmission times for the total round-trip delay.”).

182. The process for detecting such delays would have been well-known at the time of the alleged invention using, for example, an exchange mechanism (e.g. handshake) between two modules. For example, network stations could exchange probe signals and determine whether they arrive at an expected time slot. If not, delay would be introduced to compensate. Ex-1014, 15:52–16:25 and Figure 16 show a network initialization procedure that includes the exchange of a “probe” word and “probe-echo” between adjacent modules to detect the delay between them (e.g., *“to set the temporal alignment delay so that the probe words arrive synchronized to the respective routing cycles.”*):

FIG. 16 illustrates the initialization procedure 200. Initially upon power-up reset, at step 201 each channel module sends a constant "idle" bit pattern which is an out of band signal (C1). Channel modules that receive this pattern assert a signal detect bit which indicates that the channel module is coupled to another channel module and is not a non-connected port. The initialization procedure waits a programmable delay to allow stabilization of the distributed phase lock loop and clock distribution. If after synchronization at decision box 202 a channel module receives a "probe" word (C2) from a channel module to which it is coupled then at step 203 the channel module receiving the "probe" word sends a "probe-echo" (C3) word back to the probing channel module. Once the system has stabilized and if no probe word has been received by the channel module then at step 204, *the channel module will initiate the sending of a "probe" (C2) word to the channel module to which it is coupled. The "probe" (C2) word is an out of band signal*

which causes the remote channel module to send a "probe-echo" (C3) word in response. Sending of both the "probe" and "probe-echo" words are synchronized to the routing cycle of the sending channel modules. At step 205, if no signal is received from the remote channel module then the routine loops until either the receipt of the "probe-echo" word or until the receipt of a "probe" word from a remote channel module that may have sent its "probe" word at the same time. At step 206, *the receiving of either signal, "probe" or "probe-echo", provides the ability to set the temporal alignment delay so that the probe words arrive synchronized to the respective routing cycles.* If at step 207 it turns out that a channel module which sent the "probe" word receives back a "probe" word and not the "probe-echo" word, a form of deadlock in determining which of the two modules is the master for initialization purposes occurs. This case is resolved with a probabilistic protocol at step 208 where each side randomly decide whether or not to echo the C2 probe words. Once an asymmetric decision is made (via "electronically flipping a coin"), the initialization completes at step 209. The outcome of the decision is recorded in the Master/Slave bits described previously.

183. A POSA would have been motivated to implement this delay detection functionality in the SoC described by Bertozzi to provide an SoC with increased communication functionality. In particular, as I have discussed above in paragraph 179, Bertozzi recognizes the variability of the wire delays attributed to deep submicron effects and signal integrity issues, and contemplates solutions including self-synchronization techniques and on-chip networks that allow electrical

parameters of the wires to be optimized and controlled through performance optimization techniques and as such a POSA would have been motivated to implement delay detection using well-known techniques. Additionally, Bertozzi describes that its SoC is used for “audio processing” and “video processing.” Ex-1005, 2 (“Systems designed for ambient intelligence will be based on high-speed digital signal processing, with computational loads ranging from 10 MOPS for *lightweight audio processing*, 3 GOPS for *video processing*, 20 GOPS for multilingual conversation interfaces and up to 1 TOPS for synthetic video generation.”). A POSA would have understood that delay detection would enable the SoC to provide better service communications, which have precise timing constraints. Ex-1014, 8:7-9 (describing “time critical communications such as video or audio data traffic”).

184. A POSA would also have had a reasonable expectation of success including such functionality into an SoC like the one described by Bertozzi, as doing so merely combines prior art elements according to known methods to yield predictable results. As I explained earlier, utilizing “the concepts of packet switched communication from computer networks to design on-chip networks” was well-known and common at the time of the alleged invention. Ex-1006, 47. Indeed, Bertozzi itself describes doing so. Ex-1005, 5 (“The basic idea [of the NoC architecture] is borrowed from traditional large-scale multi-processors and the

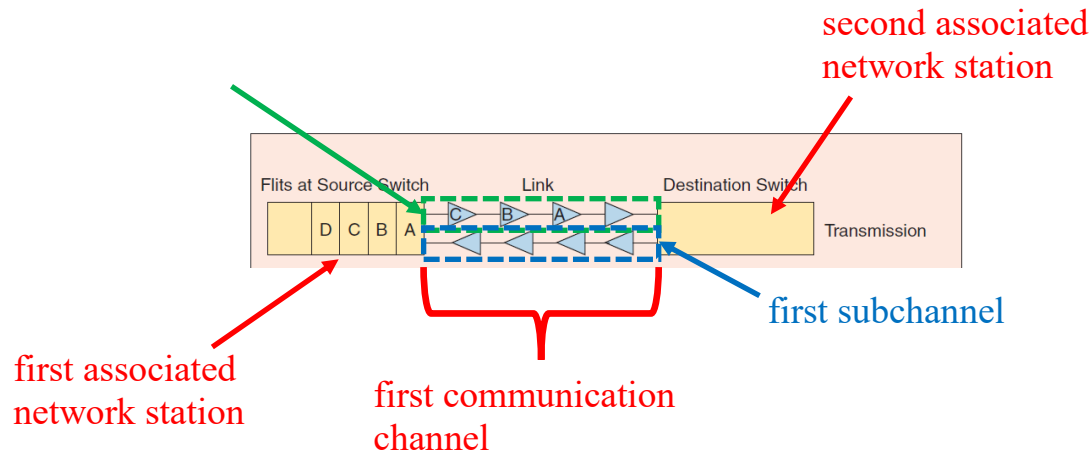
wide-area networks domain”). Further, a POSA would have understood that Bertozzi’s repeaters could be used to provide the requisite delay adjustment.

[7.g] wherein each communication channel of the plurality of communication channels comprises: a first subchannel between an input port of a first associated network station and an output port of a second associated network station; and a second subchannel between an input port of the second associated network station and an output port of the first associated network station, and

185. As discussed in Ground 1, Bertozzi discloses “subchannels.” *Supra*, [7.g]. To the extent that Patent Owner argues that each communication channel does not comprise subchannels as required by this element, this also would have been obvious in view of Bertozzi.

186. In Figure 4, Bertozzi discloses a bidirectional link with a first subchannel between an input port of a first associated network station and an output port of a second associated network station and a second subchannel between an input port of the second associated network station and an output port of the first associated network station. Ex-1005, Figs. 4, 5, 7; *supra*, Ground 1, [7.g]. I have annotated Figure 4 to show this below:

second subchannel



Ex-1005, Fig. 4 (annotations added).

187. Even if Bertozzi did not expressly or inherently teach subchannels, a POSA would have been motivated, with a reasonable expectation of success, to use bidirectional links within Bertozzi's NoC, for at least three reasons. **First**, a POSA would have known that bidirectional links allow request and response messages to be sent along the same path, which can be more efficient for sending response messages. **Second**, a POSA would have known that bidirectional links can reduce congestion throughout the network and make it less likely that a deadlock will develop. **Third**, a POSA would have known that bidirectional links allow for acknowledgements to be sent between network stations. Ex-1005, 7-8, Fig. 4.

188. I also note that in view of Bertozzi's teachings, in particular Figures 5 and 7, a POSA would have been motivated to connect the paired input and output ports of one network station to the paired input and output ports of a second

network station via the same link. Ex-1005, Figs. 5, 7; *supra*, Ground 1 [7.g]. This is at least because Bertozzi, as discussed above, expressly teaches the need for an acknowledgement (ACK) to be sent back from the destination to the source via the same path.

189. A POSA would have had a reasonable expectation of success in modifying Bertozzi’s NoC to use bidirectional links. Bidirectional links were known in the art. Ex-1005, 7-8, Fig. 4; Ex-1010, Fig. 2. As I explained above, Bertozzi discloses bidirectional links and only bidirectional links. Ex-1005, 7-8, Fig. 4. Additionally, the network stations are shown as having pairs of input/output ports for connection with bidirectional links. Ex-1005, Figs. 5, 7; Ground 1 [7.g]. Therefore, a POSA would have expected to be able to predictably to predictably implement bidirectional links, using only ordinary skill.

XIII. GROUND 3: DIELISSSEN IN VIEW OF HASSOUN RENDERS CLAIMS 1-3, 6-7, 9-10 OBVIOUS

190. [10.pre]

Claim language	Evidence and disclosure
[10.pre] A method of designing an integrated circuit comprising	Dielissen discloses this element. Dielissen discloses a method for designing an integrated circuit. <i>E.g.</i>, • Dielissen teaches the use of the “Aethereal network on chip (NoC).” Ex-1007, 1. A POSA would understand that in designing an SoC, an Aethereal NoC could be used to “integrat[e] IPs

with diverse communication requirements.” Ex-1007, 1 (“*SoC communication infrastructures, such as the $\text{\AE}thereal$ network on chip (NoC), will play a central role in integrating IPs with diverse communication requirements.*”).

- Dielissen teaches that various features “can be plugged in or left out at design time according to the requirements.” Ex-1007, 4 (“For more complex type of connections, such as narrowcast or multicast, and to provide conversions to other protocols, we add shells around the NI kernel. As an example, in Figure 7, we show a network interface with two DTL and two AXI ports. All ports provide peer-to-peer connections. In addition to this, the two DTL ports provide narrowcast connections, and one DTL and one AXI port provide multicast connections. *Note that these shells add specific functionality, and can be plugged in or left out at design time according to the requirements.*”)

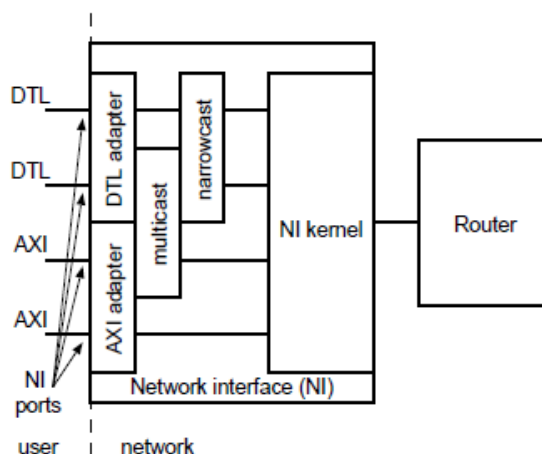


Figure 7. NI kernel and shells

- Dielissen also teaches a number of design considerations. Ex-1007, 1 (“To achieve a compositional and predictable system design, it is essential to reduce uncertainties in the

	interconnect, such as throughput and latency.”); 1 (“NoCs offer well-defined interfaces [2, 8, 14, 16], decoupling computation from communication, and easing design. It has been shown that <i>NoCs can provide interfaces to existing on-chip communication protocols</i>, such as AXI [1], OCP [12], DTL [13], thus, enabling reuse of existing IP modules [8, 15].”); 1 (“In the Æthereal NoC, we advocate the use of differentiated services and the use of guaranteed communication to eliminate uncertainties in the interconnect, and to ease integration [8]. We allow differentiated services by offering communication services on connections that can be configured individually for different services. Examples of properties that can be configured on a connection are throughput and latency that can be configured to have no guarantees (i.e., best effort) or guaranteed for a particular bound. By providing guarantees, our NoC offers predictable communication, which is a first step in designing a predictable system.”); 1 (“In the next section, we present the Æthereal NoC, which offers both guaranteed and best-effort services. The NoC consists of routers and network interfaces. Our routers, described in Section 2.1, use input queuing, wormhole routing, link-level flow control and source routing. It has two traffic classes for the GT and BE data. For GT, time slots are reserved such that no contention occurs, while for BE, we use a round-robin arbitration to solve contention. The network interfaces, described in Section 2.2, have a modular design, composed of kernel and shells. The NI kernel provides the basic functionality, including arbitration between connections, ordering, end-to-end flow control, packetization, and a link protocol with the router. Shells implement (a) additional functionality, such
--	---

	as multicast and narrowcast connections, and (b) adaptors to existing protocols, such as AXI or DTL. All these shells can be plugged in or left out at instantiation time according to the needs to optimize area cost.”); 3 (“We synthesized and layouted in a 0.13μm technology a prototype router with 6 bidirectional ports, and BE input queues of 32-bit wide and 24-word deep each (see Figure 5). In the floorplan the area-efficient custom-made hardware fifos, that we use for the BE and GT queues, are clearly visible. The design is fully testable using the well known scan-chain test method, and power stipes are included. The total area of the router sums up to 0.175mm². The router runs at a frequency of 500 MHz, and delivers a bandwidth of 16 Gbit/s per link in each direction.”); 3 (“The network interface (NI) is the component that provides the conversion of the packet-based communication of the network to the higher-level protocol that IP modules use. We split the design of the network interface in two parts: (a) the NI kernel, which packetizes messages and schedules them to the routers, implements the end-to-end flow control, and the clock domain crossing, and (b) the NI shells, which implement the connections (e.g., narrowcast, multicast), transaction ordering, and other higher-level issues specific to the protocol offered to the IP.”). A POSA would understand that SoCs are a type of integrated circuit. <i>See</i>, for example, Ex-1001, 1:8-13.
--	---

191. [1.pre]/[7.pre]

Claim language	Evidence and disclosure
[1.pre]/[7.pre] An integrated circuit comprising	Dielissen discloses this element. Dielissen discloses an integrated circuit. <i>E.g.</i>, • Dielissen teaches the use of the “Aethereal network on chip (NoC).” Ex-1007, 1. A POSA would understand that in designing an SoC, an Aethereal NoC could be used to <i>“integrat[e] IPs with diverse communication requirements.”</i> Ex-1007, 1 (<i>“SoC communication infrastructures, such as the Aethereal network on chip (NoC), will play a central role in integrating IPs with diverse communication requirements.”</i>). • Ex-1007, 1 (“As systems on chip (SoC) grow in complexity, the traditional on-chip interconnects, such as buses and switches, cannot be used anymore, due to their limited scalability. <i>Networks on chip (NoC) scale better, and, therefore, they are a solution to large SoCs</i> [2–5, 7, 9–11, 14].”). • Ex-1007, 3 (“We synthesized and layouted in a 0.13µm technology a prototype router with 6 bidirectional ports, and BE input queues of 32-bit wide and 24-word deep each (see Figure 5). In the floorplan the area-efficient custom-made hardware fifos, that we use for the BE and GT queues, are clearly visible. The design is fully testable using the well known scan-chain test method, and power stipes are included. The total area of the router sums up to 0.175mm². The router runs at a frequency of 500 MHz, and delivers a bandwidth of 16 Gbit/s per link in each direction.”)

	<i>See also</i> [10.pre]. A POSA would understand that SoCs are a type of integrated circuit. See, for example, Ex-1001, 1:8-13.
--	--

192. [10.pre.a]/[1.a]/[7.a]

Claim language	Evidence and disclosure
[10.pre.a]/[1.a]/[7.a] a plurality of functional blocks, and	Dielissen discloses this element. Dielissen discloses a plurality of IP Modules sometimes referred to as masters and slaves (“functional blocks”). <i>E.g.</i>, • Ex-1007, 1 (“The network interfaces also • implement adapters to existing on-chip protocols, such as AXI, OCP and DTL, to seamlessly connect existing IP modules to the NoC.”). • Ex-1007, 1 (“NoCs offer well-defined interfaces [2, 8, 14, 16], decoupling computation from communication, and easing design. It has been shown that NoCs can provide interfaces to existing on-chip communication protocols, such as AXI [1], OCP [12], DTL [13], thus, enabling reuse of existing IP modules [8, 15].”). • Ex-1007, 1 (“The network on chip as is exemplified by Figure 1, consists of two components: the routers and the network interfaces (NI). The routers can be randomly connected amongst themselves and to the network interfaces (i.e., there are no topology constraints). Note that in principle there can be multiple links between routers. The routers transport packets of data from one NI to another. The NIs are responsible for packetization/depacketization, for implementing the connections and services, and

for offering a standard interface (e.g., AXI or OCP) to the IP modules connected to the NoC.”).

- Ex-1007, 3 (“The network interface (NI) is the component that provides the conversion of the packet-based communication of the network to the higher-level protocol that *IP modules* use.”).
- Ex-1007, 3 (“The NI kernel (see Figure 6) receives and provides messages, which contain the data provided by the IP modules via their protocol after sequentialization. The message structure may vary depending on the protocol used by the IP module.”).
- Ex-1007, 3 (“For each connection, there are two message queues (one source queue, for messages going to the network, and one destination queue, for messages coming from the network) in the NI kernel. Their size is also selected at the NI instantiation time. Queues provide the clock domain crossing between the network and the IP modules. Each port can, therefore, have a different frequency.”).
- Ex-1007, 4 (“2.2.2 NI Shells: The interface to the IP”)

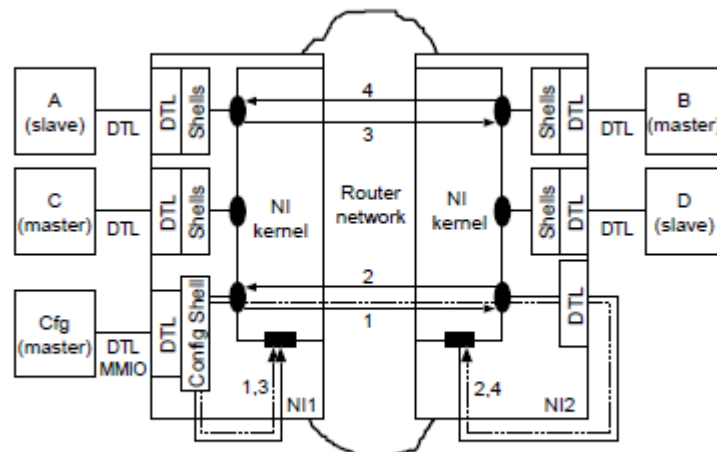


Figure 11. NI configuration

193. I have annotated Figure 11 of Dielissen to show the plurality of masters and slaves (“functional blocks”) below:

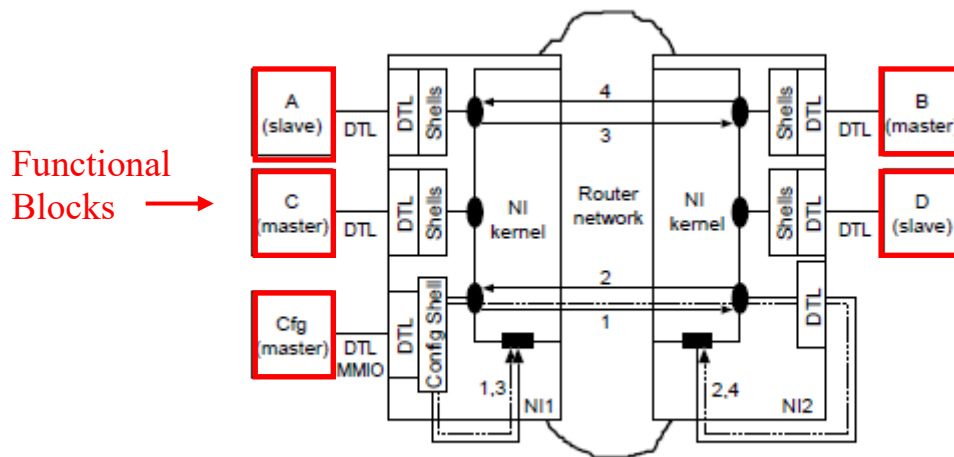


Figure 11. NI configuration

Ex-1007, Fig. 11 (red annotations added).

194. [10.pre.b]/[1.b]/[7.b]

Claim language	Evidence and disclosure
[10.pre.b]/[1.b]/[7.b] a data communication network comprising a plurality of network stations being interconnected via a plurality of communication channels for communicating data packages between the functional blocks,	Dielissen discloses this elements. Dielissen discloses a NoC (“data communication network”) comprising a plurality of network interfaces and routers (“network stations”) being interconnect via a plurality of links (“communication channels”) for communicating “data packages” (Dielissen’s flits or packets, as explained further in [10.pre.c]) between the IP modules (“functional blocks”). E.g., Ex-1007, 1 (“The network on chip as is exemplified by Figure 1, consists of two components: <i>the routers and the network interfaces (NI)</i>. <i>The routers can be randomly connected amongst themselves and to the network interfaces (i.e., there are no topology constraints)</i>). Note that in principle there can be

	multiple links between routers. The <i>routers transport packets of data from one NI to another.</i> The NIs are responsible for packetization/depacketization, for implementing the connections and services, and for offering a standard interface (e.g., AXI or OCP) to the IP modules connected to the NoC.”); Ex-1007, 1 (“SoC communication infrastructures, such as the <i>Æthereal network on chip (NoC), will play a central role in integrating IPs</i> with diverse communication requirements. To achieve a compositional and predictable system design, it is essential to reduce uncertainties in the interconnect, such as throughput and latency. In our NoC, these uncertainties are eliminated by providing guaranteed throughput and latency services. <i>Our NoC consists of routers and network interfaces. The routers provide reliable data transfer.</i> The network interfaces implement, via connections, high-level services, such as transaction ordering, throughput and latency guarantees, and end-to-end flow control. <i>The network interfaces also implement adapters to existing on-chip protocols, such as AXI, OCP and DTL, to seamlessly connect existing IP modules to the NoC.</i> These services are implemented in hardware to achieve high speed, and low area.”) • Ex-1007, 2 (“<i>Routers send data from one network interface to the other by means of packets. Such a packet consists of one or more flits, where a flit is the minimal transmission unit.</i> As a transmission scheme we use wormhole routing, because of the low cost (the buffer capacity can be less than the length of a packet) and low latency (the router can start forwarding the first flit of a packet without waiting for the tail). To reduce queuing capacity of a router, and thus the area, input queuing is used, as shown in Figure 2.”).
--	--

- Ex-1007, 1 (“*The NIs are responsible for packetization/depacketization, for implementing the connections and services, and for offering a standard interface (e.g., AXI or OCP) to the IP modules connected to the NoC.*”).
- Ex-1007, Figs. 1, 11.

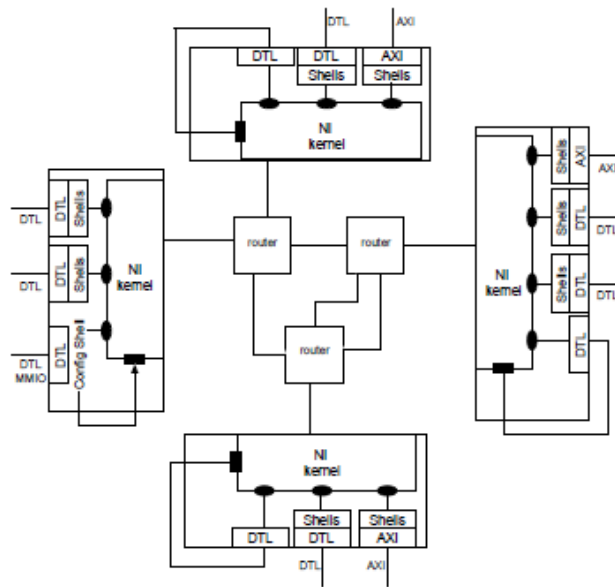


Figure 1. network example

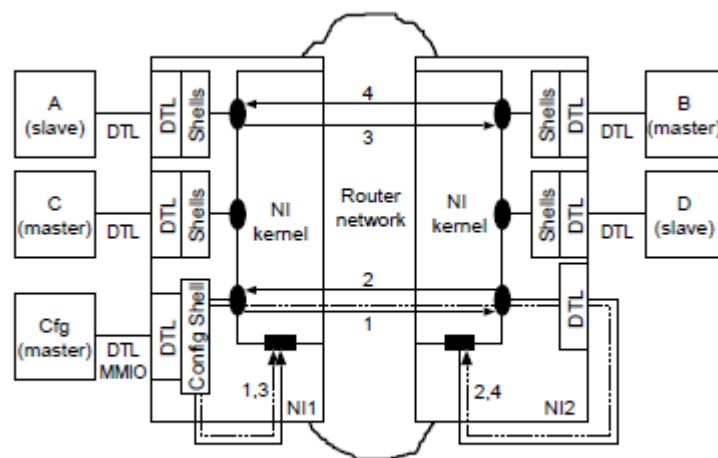


Figure 11. NI configuration

195. The NoC “consists of two components: the routers and the network interfaces (NI).” Dielissen at 1. Dielissen’s routers and network interfaces are connected by links. Ex-1007, 1 (“The routers can be randomly connected amongst themselves and to the network interfaces.... Note that in principle there can be multiple links between routers.”).

196. I have annotated these elements on Figures 1 and 11 below:

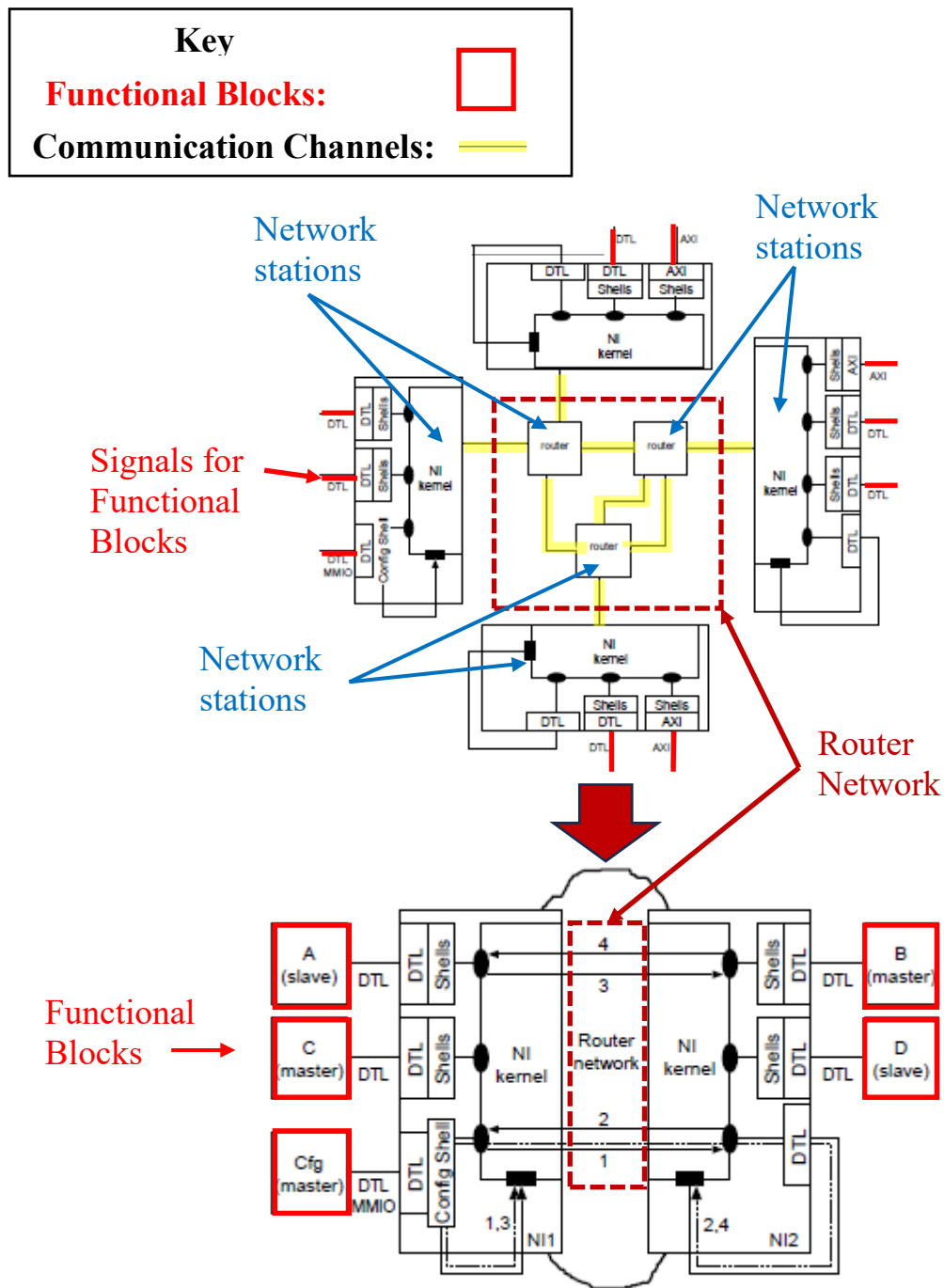
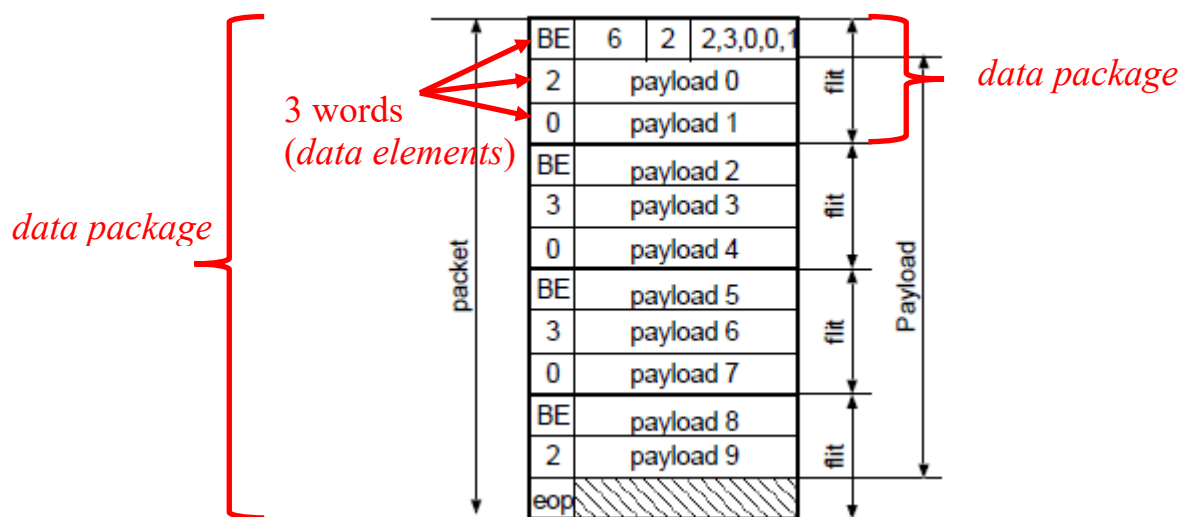


Figure 11. NI configuration

Ex-1007, Figs. 1 and 11 (annotations added).

197. The routers and NIs communicate data packages between the IP modules (*functional blocks*). Ex-1007, 2 (“Routers send data from one network

interface to the other by means of packets.”); 1 (“The NIs are responsible for packetization/depacketization, for implementing the connections and services, and for offering a standard interface (e.g., AXI or OCP) to the IP modules connected to the NoC.”). Dielissen discloses the claimed data packages via its teaching of data packets and via its teaching of flits within the data packets. Dielissen teaches that “Routers send data from one network interface to the other by means of packets” and that each packet “consist[]s of one or more flits,” and that “a flit is the minimal transmission unit.” Ex-1007, 2, Figs. 3, 4. Each of Dielissen’s packets, and each flit within the packet, contains multiple data elements and corresponds to a data package as claimed. I have illustrated this on Figure 4 of Dielissen below:



Ex-1007, Fig. 4. The packets are broken into flits. Ex-1007, 2 (“Such a packet consists of one or more flits, where [sic] a flit is the minimal transmission unit.”).¹² Dielissen discloses an implementation that each flit is set to contain 3 words (i.e., data elements). Ex-1007, p.2, Figs. 3, 4.

198. [10.pre.c]/[1.c]/[7.c]

Claim language	Evidence and disclosure
[10.pre.c]/[1.c]/[7.c] each data package comprising N data elements including a data element comprising routing information for the network stations, N being an integer of at least two,	This element is disclosed by Dielissen. Dielissen discloses that its packets are divided into flits, and that each flit contains 3 words. Ex-1007, 2 (“Besides the path, the header also contains information for the Network interface, which is explained in section 2.2.1. As explained, a network packet is build up of one or more network flits. For the current Æthereal network, the flit size is chosen 3 to optimize the data clock frequency and control frequency. The information of the type of flit is annotated in the first element of the sideband information, being the id. The format of the flits is shown in Figure 3. The figure shows that the flit contains a header and 2 payload words. Only the first flit of a packet has a header and as a consequence, the next flits can have 3 payload words. Note that when packets consist of multiple flits, the overhead of the header is reduced. The amount of valid words in the flit is stored in the size field. The end of the packet is notified by the eop flag in the sideband information. As an example, Figure 4 shows how a packet, containing 10 payload words can be build up.”).

¹² A flit is a flow instruction unit. Ex-1001, 4:57. This is the smallest amount of data that the NoC will send on an end-to-end basis. Such a flit may contain multiple phits or words, the amount of data that can be transferred across a physical channel in a single step or clock cycle. Dielissen teaches that each of its flits contains 3 words. Ex-1007, p.2, Figs. 3, 4.

	Dielissen discloses that each packet has a header that includes the path on which the packet travels and that each flit has a flit type field and an end of packet flag <i>E.g.</i>, • Ex-1007, 2 (“We select source routing as an addressing scheme, because it allows topology independence, while at the same time has a low cost: no expensive (programmable) lookup tables are needed in the router. <i>In source routing the path on which the packet travels is included in the header of a packet.</i> In <i>Æthereal</i>, this path is a list of destination ports, from which each router on the path removes the first element for its own use.”) • Ex-1007, 2 (“We include <i>the switching configuration (the path) in the packet header,</i> and, as a consequence the slot tables are removed from the routers.”) • Ex-1007, 2 (“The information of the <i>type of flit is annotated in the first element of the sideband information, being the id.</i> The format of the flits is shown in Figure 3. <i>The figure shows that the flit contains a header and 2 payload words.</i> Only the first flit of a packet has a header and as a consequence, the next flits can have 3 payload words. Note that when packets consist of multiple flits, the overhead of the header is reduced. The amount of valid words in the flit is stored in the size field. <i>The end of the packet is notified by the eop flag in the sideband information.</i> As an example, Figure 4 shows how a packet, containing 10 payload words can be build up.”) • Ex-1007, 2-3 (“The router has a controller and a data path elements. In the data path, the input messages, from either routers or network interfaces are parsed by the header-parsing units (hpu). These units, shown in Figure 2, remove the first element for the path, send the parsed flits into
--	---

GT or BE queues and notify the controller that there is a packet. The controller schedules flits for the next cycle. After scheduling the GT-flits, the remaining destination ports can serve the BE-flits. In the case of conflicts (e.g. two BE-flits address the same destination), a round-robin arbitration scheme is applied. The controller sets the switches in the right direction for the duration of the next flit cycle. Furthermore the read commands will be given to the fifo's.”)

- Ex-1007, Figs. 3, 4

id	credit	qid	path
size	payload 0		
eop	payload 1		

Figure 3. Network flits

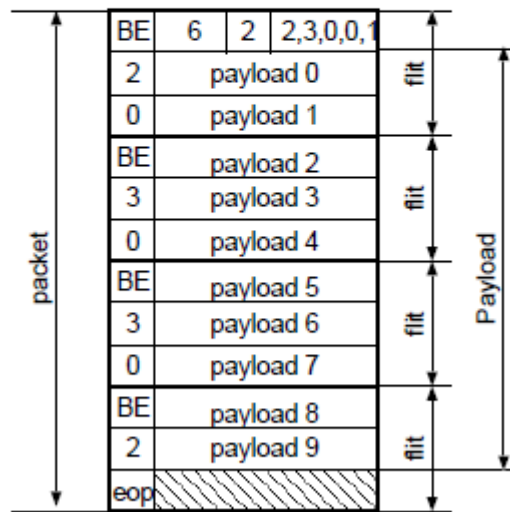


Figure 4. BE-packet example

199. The ratio of “flits” to “phits” is defined differently in Dielissen than in Bertozzi. Dielissen—like the specification of the ’2893 patent—describes an implementation where each flit contains more than one data elements (specifically,

three data elements in Dielissen). Dielissen teaches that each of its flits is defined to have three words¹³. Ex-1007, 2; Figs. 3, 4. As noted above, a POSA would recognize that a flit in this implementation corresponds to a data package.

Under the Petitioner Qualcomm's Construction

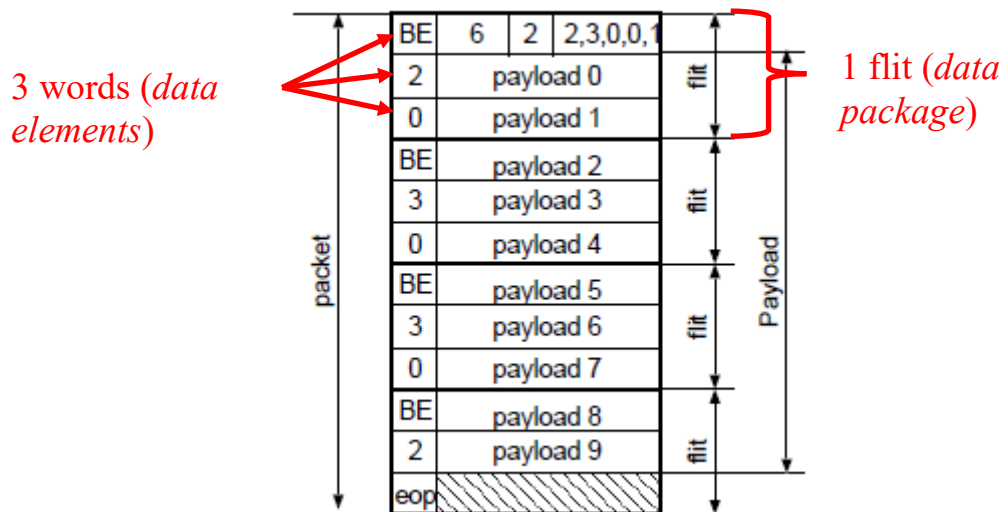
200. Petitioner Qualcomm's construction requires that "every data package communicated by the network stations has a fixed size of N data elements." Ex-1025, 32.

201. It is my opinion that under Petitioner Qualcomm's Construction, Dielissen discloses that every flit ("data package") has three words ("comprising N data elements") including a word ("data element") comprising "routing information" for the routers and NIs ("network stations"), 3 (N) being an integer of at least two.

202. As discussed in the evidence and disclosure table and shown in Figure 3 and 4, Dielissen discloses that its packets are divided into flits. Ex-1007, 2 ("Such a packet consists of one or more flits"). Each flit in the Aethereal system taught by Dielissen contains 3 words. Ex-1007, 2 ("the flit size is chosen 3"). A

¹³ Dielissen's use of "word" is equivalent to Bertozzi's use of phit (and Bertozzi's flit in its embodiments) such that it means the amount of data transferred over one clock cycle.

word, which a POSA would have understood corresponds to the claimed “data element,” is the unit of information that can be transferred across a physical channel in a single clock cycle. Below, I have annotated Figure 4 to which shows four flits, each containing three words:



Ex-1007, Fig. 4 (red annotation added).

203. It is my opinion that each flit includes a word (“data element”) comprising routing information for the routers and network interfaces. Dielissen discloses a data element comprising routing information for the network stations in the same way as the ’2893 patent. The ’2893 patent indicates that routing information includes “destination and required service type (e.g. best effort or guaranteed bandwidth). . . .” Ex-1001, 2:22-24. In the context of the ’2893 patent as taught by its specification, the first data package of a stream will contain a header. Ex-1001, 5:29-34. This header will contain “*information about the destination of*

the data package as well as service level of the data package, e.g. guaranteed bandwidth or best effort service.” *Id.* After a header is received, “[t]he service level will be flagged by setting a bit in the appropriate memory element of the destination output port....” Ex-1001, 5:32-34. For subsequent data packages of the same stream, the service level is identified and “the data router will know that these data packages belong to” the same data stream that is in progress. Ex-1001, 5:34-38. The data packages also have an indicator to tell the router if the current data package is the last of the data stream. Ex-1001, 5:38-40.

Typically, *the header of a first data package will comprise information about the destination of the data package as well as the service level of the data package*, e.g. guaranteed bandwidth or best effort service. The service level will be flagged by setting a bit in the appropriate memory element of the destination output port of the data router. For *subsequent data packages having the same service level, the data router will know that these data packages belong to a data stream in progress* because the appropriate bit has been stored in the associated memory element. The *last data package of the data stream contains an indicator causing the data router to reset said bit*. Such a mechanism *allows interleaving of data packages of two data streams over a single communication channel*.

Ex-1001, 5:29-42.

204. Dielissen provides a similar disclosure. Dielissen discloses that the first word of a packet is a packet header that contains the routing path, the packet id, and the flit type. Ex-1007, 2-3, Figs. 3, 4. Subsequent flits of the packet contain a flit type field that indicates whether the flit is a guaranteed throughput or best

effort and an end of packet (“eop”) flag in the sideband information.¹⁴ Ex-1007, 2-

3. The flit type field is like the service type identified in the ’2893 patent and the

¹⁴ I do not agree that sideband signals are considered to be part of the data package or that such signals can be routing information. However, Patent Owner has identified analogous signals in its infringement contentions and I have been asked to apply that interpretation in this proceeding. Ex-1019, 14-19; *see, e.g., id.*, 16-17:

11.3.1.3 Physical Layer

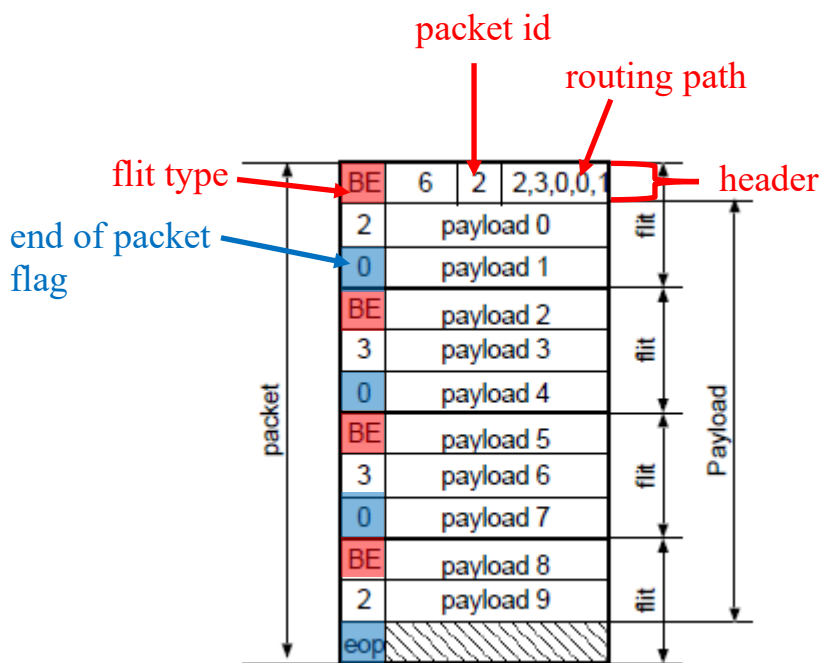
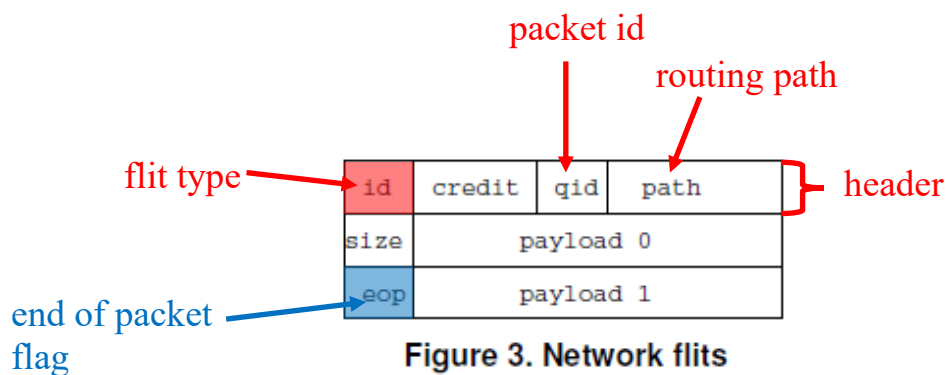
The delivery of packets within the NoC is the responsibility of the physical layer. Packets, which have been split by the transport layer into cells, are delivered as words that are sent along links. Within a single clock cycle, the physical layer may carry words comprising a fraction of a cell, a single cell, or multiple cells. The link size, or width (i.e., number of wires), is set by the designer at design time and determines the number of cells of one word. NTTP defines five possible link-widths: quarter (QRT), half (HLF), single (SGL), double (DBL), and quad (QUAD). A single-width (SGL) link transmits one cell per clock cycle, a double-width link transmits two cells per clock cycle, and so on. Words travel within point-to-point links, which are independent from other protocol layers: a word is sent through a transmit port, Tx, over a link to a receive port, Rx. The actual number of wires in a link depends on the

eop flag is like the last data package indicator. *Compare* Ex-1007, 2-3 with Ex-1001, 5:29-40. Therefore, a POSA would have recognized that each of Dielissen's data packages (in the form of flits) contain routing information in the form of the flit type field and the eop flag.

205. I have annotated Figures 3 and 4 to show these data fields below:

maximum cell-width (header, necker, and data cell) and the link-width. One link (represented in [Figure 11.1](#)) defines the following signals:

- **Data**—Data word of the width specified at design-time.
- **Frm**—When asserted high, indicates that a packet is being transmitted.
- **Head**—When asserted high, indicates the current word contains a packet header. When the link-width is smaller than single (SGL), the header transmission is split into several word transfers. However, the Head signal is asserted during the first transfer only.
- **TailOfs**—Packet tail: when asserted high, indicates that the current word contains the last packet cell. When the link-width is smaller than single (SGL), the last cell transmission is split into several word transfers. However, the Tail signal is asserted during the first transfer only.
- **Pres.**—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in [Figure 11.2](#)).
- **Vld**—Data valid: when asserted high, indicates that a word is being transmitted.
- **RxRdy**—Flow control: when asserted high, the receiver is ready to accept word. When de-asserted, the receiver is busy.



Ex-1007, Figs. 3, 4 (red and blue annotations added).

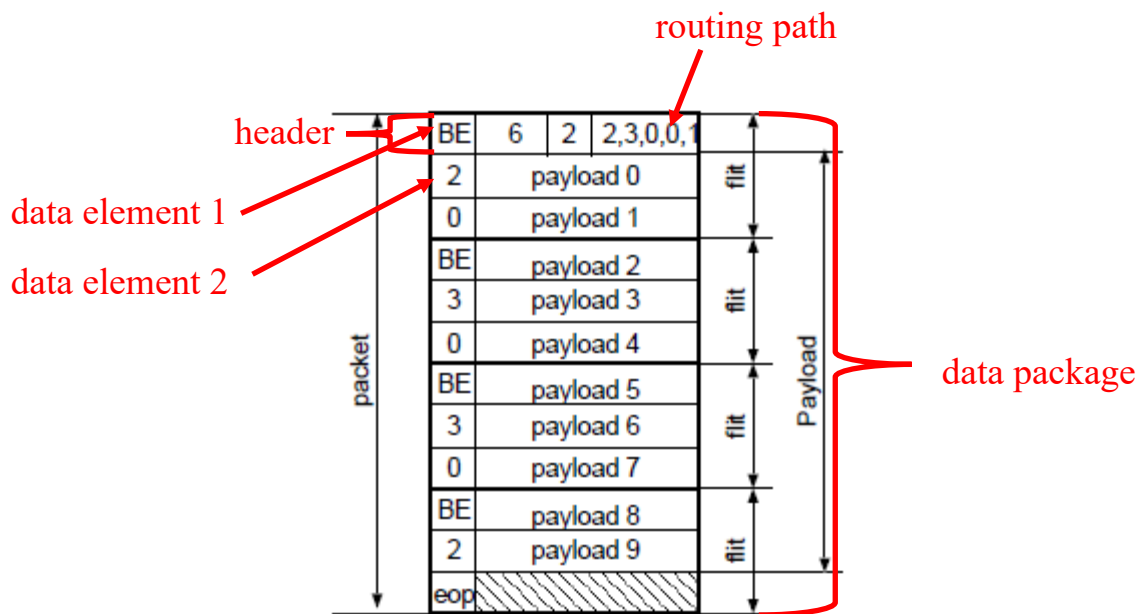
Under Patent Owner's Construction

206. Dielissen also discloses this element under Patent Owner's construction. Under Patent Owner's interpretation that the claim only requires each data package to have "at least N data elements" (*see supra*, §VIII).

207. It is my opinion that under Patent Owner's Construction, Dielissen discloses this element for the same reasons as discussed above under Petitioner Qualcomm's construction. Alternatively, it is my opinion that under Patent Owner's broader construction, Dielissen also discloses that every data packet ("data package") comprises a flit with at least 2 words ("data elements") including a header word ("data element") comprising routing information for the routers and NIs ("network stations"), 2 being an integer N of at least two.

208. As discussed above in the evidence and disclosure table, Dielissen discloses that its packets are divided into flits. Ex-1007, 2 ("a network packet is build [sic] up of one or more network flits."). Each flit in the Aethereal system taught by Dielissen contains three words. Ex-1007, 2 ("the flit size is chosen 3"). As each packet has at least one flit, under Patent Owner's construction, each packet ("data package") comprises $N=2$ words ("data elements").¹⁵ I have illustrated this below on Dielissen Fig. 4:

¹⁵ I disagree with Patent Owner's construction of "at least N data elements" at least because it would allow for this unbounded ability to interpret what a data package is, as illustrated in the example herein where simply setting $N=2$, an integer, would result in such a nonsensical example since the '2893 specification is clear that data packages have uniform, equal size (fixed size) as I explained previously.



Ex-1007, Fig. 4 (red annotation added). Each packet (“data package”) includes a header word (“data element”) comprising routing information for the routers and NIs. As discussed above in the evidence and disclosure table, Dielissen discloses that the first word of a packet is a packet header that contains the routing path. Ex-1007, 2-3, Fig. 3, 4. A POSA would understand that the routing path would be routing information. For example, Dielissen teaches that this information includes a “list of destination ports” that are used by the routers. Ex-1007, 2.

209. [10.pre.d]/[1.d]/[7.d]

Claim language	Evidence and disclosure
[10.pre.d]/[1.d]/[7.d] the plurality of network stations comprising a plurality of data routers and a plurality of network	Dielissen discloses this element. Dielissen’s NoC comprises a plurality of network stations, including a plurality of routers (“data routers”) and a plurality of network interfaces (“network interfaces”), and each of the routers (“data routers”) is

interfaces, each of the data routers being coupled to a functional block via a network interface;	coupled to an IP module (“functional block”) via a network interface (“network interface”). E.g., • Ex-1007, 1 (“<i>The network on chip as is exemplified by Figure 1, consists of two components: the routers and the network interfaces (NI).</i> The routers can be randomly connected amongst themselves and to the network interfaces (i.e., there are no topology constraints). Note that in principle there can be multiple links between routers. The routers transport packets of data from one NI to another. <i>The NIs are responsible for packetization/depacketization, for implementing the connections and services, and for offering a standard interface (e.g., AXI or OCP) to the IP modules connected to the NoC.</i>”) • Ex-1007, 1 (“SoC communication infrastructures, such as the Æthereal network on chip (NoC), will play a central role in integrating IPs with diverse communication requirements. To achieve a compositional and predictable system design, it is essential to reduce uncertainties in the interconnect, such as throughput and latency. In our NoC, these uncertainties are eliminated by providing guaranteed throughput and latency services. <i>Our NoC consists of routers and network interfaces. The routers provide reliable data transfer.</i> The network interfaces implement, via connections, high-level services, such as transaction ordering, throughput and latency guarantees, and end-to-end flow control. <i>The network interfaces also implement adapters to existing on-chip protocols, such as AXI, OCP and DTL, to seamlessly connect existing IP modules to the NoC.</i> These services are implemented in hardware to achieve high speed, and low area.”) • Ex-1007, 2 (“<i>Routers send data from one network interface to the other by means of packets.</i> Such a packet consists of one or more
--	--

	flits, where a flit is the minimal transmission unit. As a transmission scheme we use wormhole routing, because of the low cost (the buffer capacity can be less than the length of a packet) and low latency (the router can start forwarding the first flit of a packet without waiting for the tail). To reduce queuing capacity of a router, and thus the area, input queuing is used, as shown in Figure 2.”) • Ex-1007, 3 (“<i>The network interface (NI) is the component that provides the conversion of the packet-based communication of the network to the higher-level protocol that IP modules use.</i> We split the design of the network interface in two parts: (a) the NI kernel, which packetizes messages and schedules them to the routers, implements the end-to-end flow control, and the clock domain crossing, and (b) the NI shells, which implement the connections (e.g., narrowcast, multicast), transaction ordering, and other higher-level issues specific to the protocol offered to the IP. We describe the architectures of the NI kernel and the NI shells in the next two sections, and the results for their implementation in Section 2.2.3.”). • Ex-1007, 4 (“2.2.2 NI Shells: The interface to the IP”) • Ex-1007, 4 (“In Figures 8 and 9, we show a master and slave shells that implement a simplified version of a protocol such as AXI. The basic functionality of such a shell is to sequentialize [sic] commands and their flags, addresses, and write data in request messages, and to desequentialize [sic] messages into read data, and write responses. Examples of the message structures (i.e., after sequentialization) passing from NI shells and NI kernel are shown in Figure 10.”) • Ex-1007, Figs. 1, 8, 9, 11.
--	--

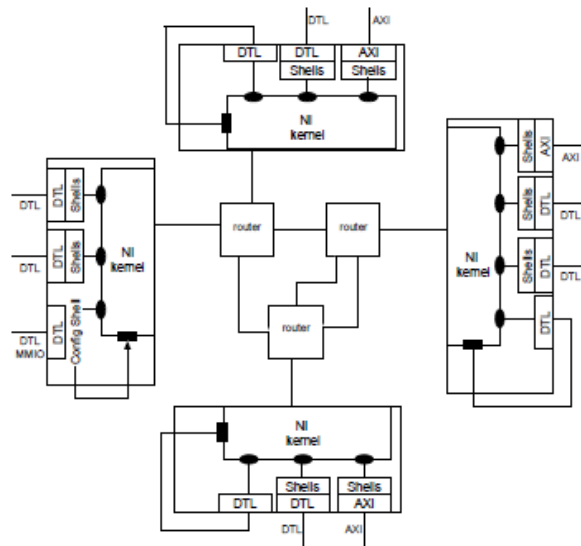


Figure 1. network example

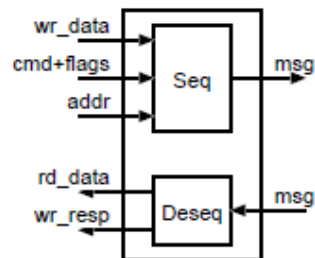


Figure 8. Master shell

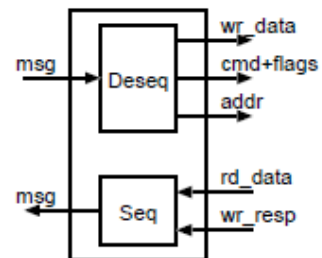


Figure 9. Slave shell

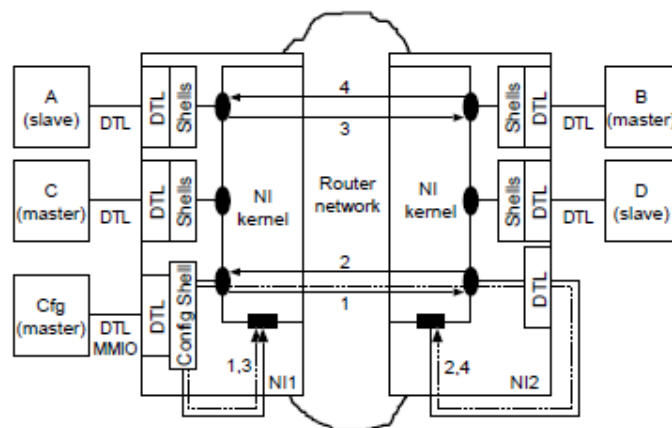


Figure 11. NI configuration

210. Figures 1 and 11 show that the routers are coupled to the IP modules via NIs (network interfaces). A POSA would have understood that the signals on the outside of the NIs (network interfaces) in Figure 1 are to be connected to IP modules. Ex-1007, 1 (“The NIs are responsible for packetization/depacketization, for implementing the connections and services, and for *offering a standard interface (e.g., AXI or OCP) to the IP modules* connected to the NoC.”). For example, Figure 1 shows that the NIs have DTL and AXI connection. Ex-1007, Fig. 1. A POSA would understand that these would be connected to IP modules that communicate with those protocols. Ex-1007, 1 (“The *network interfaces also implement adapters to existing on-chip protocols, such as AXI, OCP and DTL, to seamlessly connect existing IP modules to the NoC.*”). A POSA would also have understood and at once envisaged that the “Router network” of Figure 11 would be made up of routers like those in Figure 1. I have annotated these elements below on Figures 1 and 11:

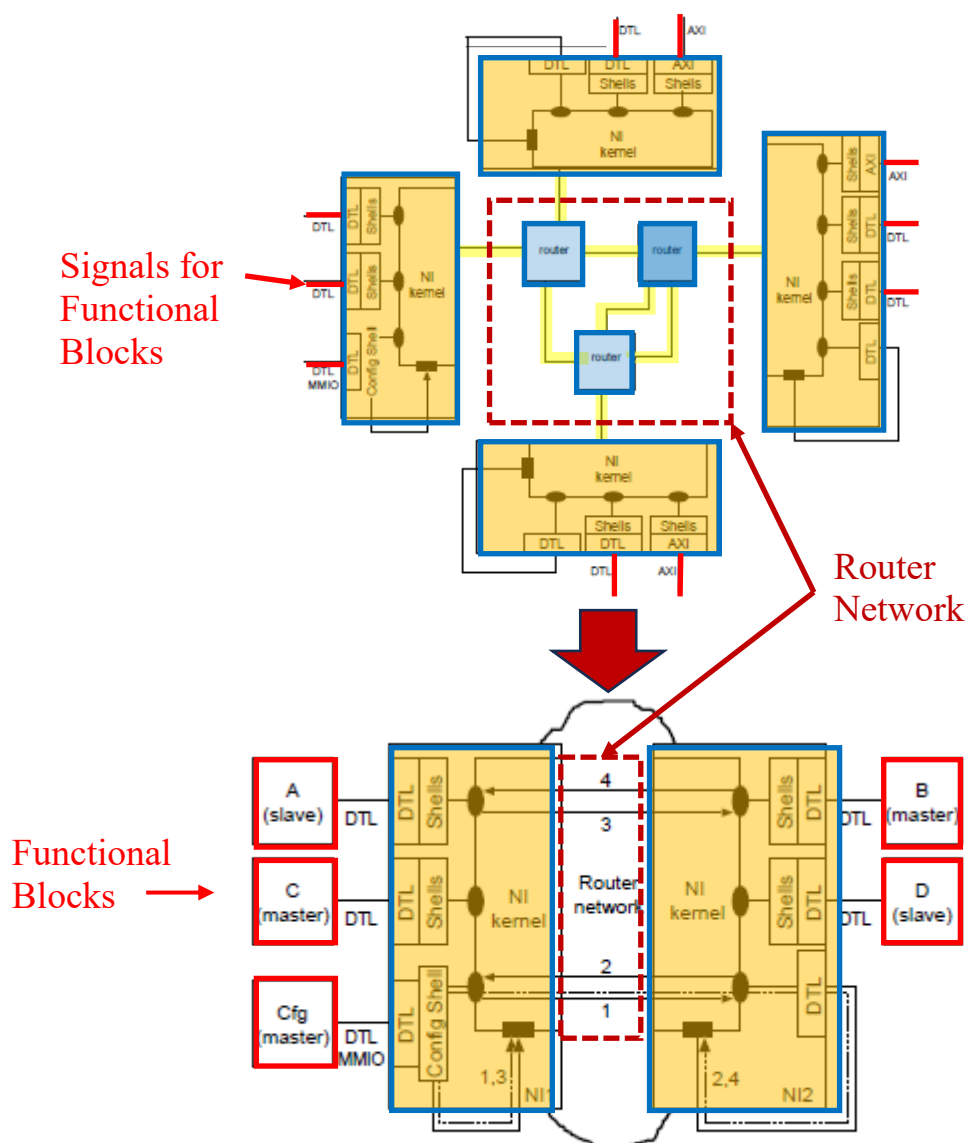
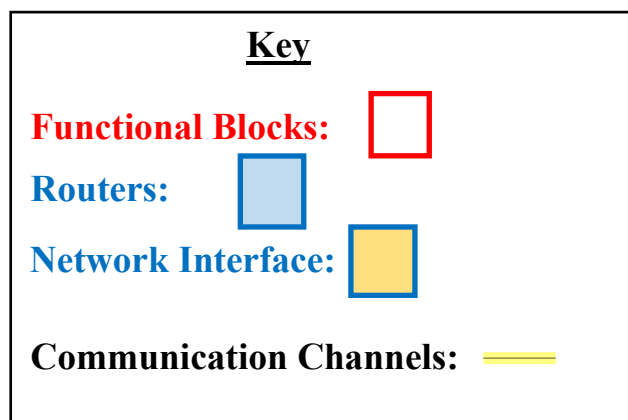


Figure 11. NI configuration

211. [10.e] and [10.f]

Claim language	Evidence and disclosure
[10.e] the method comprising the acts of: identifying a first communication channel between a first network station and a second network station that has a data transfer delay exceeding a predefined delay threshold; and [10.f] in response to the identifying act, inserting M*N data storage elements into the data communication network, M being a positive integer, for introducing a delay of M*N cycles on the first communication channel.	Dielissen does not disclose these elements. However, Dielissen in view of Hassoun teaches these limitations. Hassoun teaches identifying a link (“<i>first communication channel</i>”) between a source (“<i>first network station</i>”) and a destination (“<i>second network station</i>”) that has a data transfer delay exceeding a predefined delay threshold. For example: • Ex-1011, 1 (“Shrinking process geometries and the increasing use of IP components in SoC designs give rise to new problems in routing and buffer insertion. A particular concern is that cross-chip routing will require multiple clock cycles.”) • Ex-1011, 1 (“Three distinct trends will pose new muting challenges in future SoC designs. First, SoCs will utilize several IP (Intellectual Property) components, both soft and hard, like embedded processors and memories. A shortest path for a signal between two chip components may thus be obstructed by IP blocks. Second, <i>the drive for higher performance will continue to push the clocking frequencies. Third, shrinking process geometries and improvement in process technologies allows building bigger dies. Multiple clock cycles will be required to cross a chip.</i> Furthermore, if the IPs are clocked at different frequencies, as is the case with hard IP that often has a fixed clock period, then the signal route must cross from one time domain (of the sender) to another (that of the receiver) latched through the proper circuitry. In contrast to a system with a single clock, or a single clock domain system, a system with multiple clocks is

	often referred to as a multiple clock domain system. The clocking scheme is referred to as Globally Asynchronous Locally Asynchronous [sic], or GALS [3,8].”) • Ex-1011, 1 (“This paper explores simultaneous routing and buffer insertion in the context of single and multiple clock domains.”) • Ex-1011, 1 (“The objective is to minimize the cycle latency, or equivalently, the total number of registers along the route.”) • Ex-1011, 2-3 (“We now explore the problem of finding a buffered routing path from s to t <i>when multiple clock cycles are required. Routing over large distances in increasingly aggressive technologies will require several clock cycles to cross the die. Hence, one must periodically clock the signal by inserting synchronization elements (such as registers) along the signal path.</i> In this case, one cannot simply treat a register like a buffer and add the register delay to the existing path delay in the Fast Path algorithm. <i>The realizable delay between consecutive registers on a path will always be determined by the clock period, regardless of the actual signal propagation time. Register-to-register sub-paths with delays larger than the permissible clock cycle are not permitted.</i>”) • Ex-1011, 3 (“Let r denote the register to be used for insertion, T_ϕ the clock period, and $Setup(r)$ to be the setup time for r. We extend the definition of node labeling to permit register assignment, i.e., $m(v) = r$ for any node $v \in V - \{s, t\}$. We also assume that the source and sink are synchronization elements, so that $g_s = g_t = r$. We define the clock period constraint as follows: a buffer-register path is feasible if and only if $p(v) = I$ whenever $m(v) \in I$ and the buffered path delay between consecutive registers is less than or equal to $T_\phi - Setup(r)$.”)
--	---

- Ex-1011, 3 (“Figure 2 shows an example of a buffered-register path on a grid graph with both circuit and wire blockages. ***Because a register releases its signal with clock cycle, the s-t path latency is given by $T_{\phi} \times (p + 1)$*** , where p is the number of registers on the s-t path. ***The s-t path has three registers between s and t. It thus takes four clock cycles to traverse from s to t despite the different spacings between consecutive registers.***”)
- Ex-1011, Fig. 2

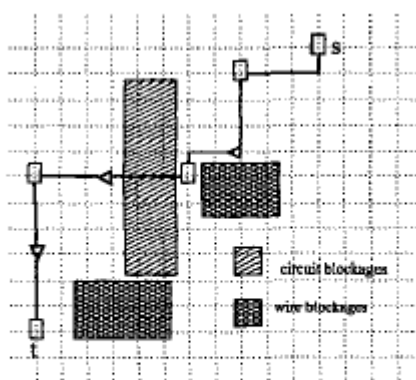


Figure 2. An example of routing within a single clock domain

Hassoun teaches that it will take “[M]ultiple clock cycles...to cross a chip.” Ex-1011, 1. Hassoun teaches to “periodically clock” such signals by inserting registers along the signal path. Ex-1011, 1-2. Hassoun teaches that sub-path, the paths between the registers, cannot have delays greater than the clock period. Ex-1011, 2 (“Register-to-register sub-paths with ***delays larger than the permissible clock cycle are not permitted.***”). Therefore, a POSA would understand that in order to compare a sub-path delay (the claim’s “data transfer delay”) to a desired clock rate (i.e., frequency) in order to determine if a length does or does not satisfy a predefined timing requirement, the comparison has to be

	made to a delay threshold that represents the desired clock frequency of the design. As a result of that comparison, wires are partitioned via repeaters (e.g., relay stations or latches) in order to satisfy the timing requirement. <i>See also</i> Ground 1, [10.e] and [10.f].
--	--

212. Global wires are long wires that communicate signals across a chip. For example, a global wire could connect a pair of functional blocks on opposite sides of a chip. Dealing with delays caused by global wires or other long wires was a known issue in on-chip interconnects. *E.g.*, Ex-1005, 3 (“While gate delays scale down with technology, ***global wire delays typically increase*** or remain constant ***as repeaters are inserted***). It is estimated that in 50 nm technology, at a clock frequency of 10 GHz, a global wire delay can be up to 6–10 clock cycles.”), 6 (“As ***geometries shrink***, gate delay improves much faster than the delay in long wires. Therefore, the ***long wires increasingly determine the maximum clock rate***, and hence performance, of the entire design.”); Ex-1010, 1 (“Links ***can be pipelined with a flexible number of stages to decouple data introduction speed from worst-case link delay***.”); Ex-1011, 1 (“Multiple clock cycles will be required to cross a chip.”); Ex-1006, 36 (“The physical communication between intellectual property blocks (IP) has made use of a mix of local and global wires.”).

213. Long wires were known to limit the performance of a chip by causing a data transfer delay exceeding a predefined threshold, such as exceeding the time

allocated for a single clock cycle. Ex-1005, 3, 6; Ex-1010, 1, 2. Logic gates, one of the basic building blocks of SoCs, have gotten faster as transistors have become smaller. At the same time, wire delays have remained almost constant. This has resulted in a disparity between gate delay and wire delays¹⁶. Ex-1005, 6 (“As *geometries shrink, gate delay improves much faster than the delay in long wires.*”); Ex-1006, 36 (“several deep-submicron problems must be solved: the increasing *disparity between transistor and wire speeds*”). The result of this disparity is that “*long wires increasingly determine the maximum clock rate*, and hence performance, of the entire chip.” Ex-1005, 6. A POSA would have recognized this as a problem (and at least as a potential problem) with the Aethereal NoC, which is described in Dielissen. Ex-1007, 1 (“we present the Aethereal NoC”).

214. For example, other papers by the same authors discussing the Aethereal NoC, mention global wires. Ex-1012, 1 (“Networks on a chip (NoC) are emerging as an alternative to existing on-chip interconnects because they...structure and manage *global wires* in new deep-submicron technologies”) and 299 (“*Long wires can be pipelined*”); Ex-1006, 36 (“several deep-submicron problems must be solved: *the increasing disparity between transistor and wire*

¹⁶ Wire delay is the time that it takes a signal to propagate across a wire. Gate delay is the time that it takes for a signal to propagate through a series of logic gates.

speeds”) and (“The physical communication between intellectual property blocks (IP) has made use of a *mix of local and global wires*. When timing constraints of the design must be verified, IPs cannot be checked independently because they are interconnected; correcting a timing violation in one IP may invalidate the timing of another. This process does not necessarily converge to a solution, and is design specific (not re-usable). This global timing closure problem results from a tight (timing) coupling [3].”). Slowing down the clock as a result of a long wire can result in a chip being less effective. A POSA would understand that a chip operating at a lower clock rate would not be able to perform as many operations in a given time period compared to that same chip if it were able to operate at a higher clock rate.

215. Hassoun (and other references) teach that it was a known solution to the problem of delay caused by long or “global” wires to “pipeline” long links with repeaters. Hassoun teaches to insert repeaters when “multiple clock cycles are required,” explaining that “[r]outing over large distances in increasingly aggressive technologies will require several clock cycles to cross the die. Hence, one must periodically clock the signal by inserting synchronization elements (such as registers) along the signal path.” Ex-1011, 2-3; *see also* Ex-1005, 3, 7-8 (“A solution to overcome the interconnect-delay problem *consists of pipelining interconnects*”); Ex-1010, 1-3 (“long links can be pipelined with a flexible number

of stages to decouple data introduction speed from worst-case link delay.”); Ex-1012 (paper by the authors of Dielissen discussing option regarding the Aethereal NoC), 6 (“***Long wires can be pipelined***”).

216. This means identifying a communication channel between network stations that has a transfer delay exceeding a predefined delay threshold, and then inserting repeaters at intervals along the wire. Repeaters allow wires to be partitioned into segments that are short enough to satisfy a faster clock rate, thus allowing an increase in the clock frequency. Ex-1005, 6, 7-8 (“A solution to overcome the interconnect-delay problem consists of ***pipelining interconnects*** [34, 35]. Wires can be partitioned into segments (or ***relay stations, which have a function similar to the one of latches in a pipelined data path***) whose length satisfies pre-defined timing requirements (e.g., the desired clock speed of the design). In this way, link delay is changed into latency, but the data introduction rate is not bound by the link delay any more.”) and (“Switch-to-switch links are subdivided into basic segments whose length can be ***tailored to the desired clock frequency*** that needs to be achieved in the design.”); Ex-1010, 2 (“Our switch-to-switch link is subdivided into basic segments whose length guarantees that the desired clock frequency (i.e. the maximum speed provided by a certain technology) can be used.”); Ex-1011, 1 (“the drive for higher performance will continue to push the clocking frequencies.”), 2-3.

217. For example, Hassoun discloses wires partitioned into segments by multiple (three) repeaters/registers. I have annotated these registers below:

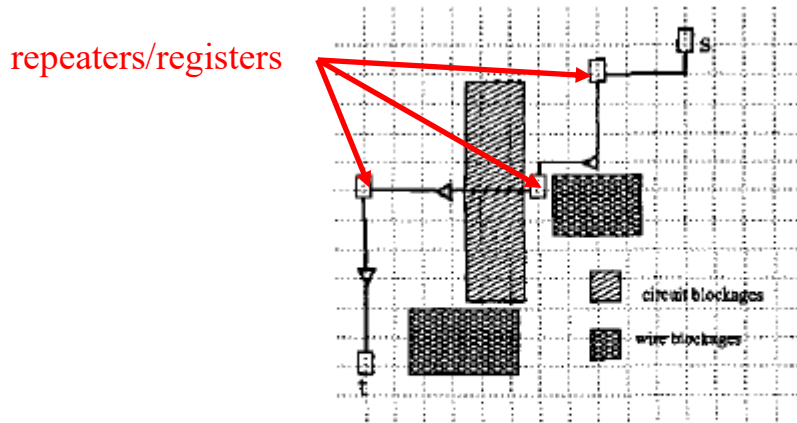
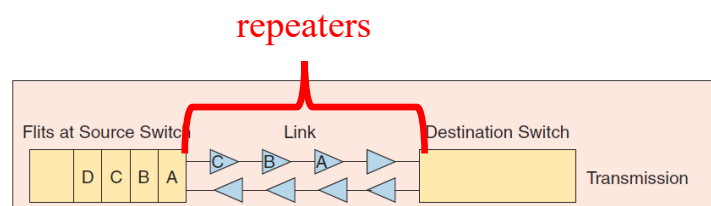
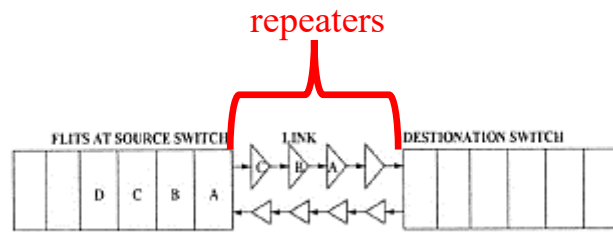


Figure 2. An example of routing within a single clock domain

Ex-1011, Fig. 2 (red annotation added). Other references similarly teach the use of repeaters on links. I have also annotated Figure 4 of Bertozzi showing four repeaters added on each direction of the link:



Ex-1005, Fig. 4 (red annotations added). Similarly, I have annotated Figure 2 of Dall'Osso which shows a link with four repeaters added on each direction of the link:



Ex-1010, Fig. 2 (red annotations added). Further, pipelining long links was discussed as an option in another paper by the authors of Dielissen regarding the Aethereal NoC. Ex-1012, 3 (“*long wires can be pipelined*”).

218. A POSA would expect pipelining long wires to allow for an improved clock rate used by a chip. Because the wire segments that result from the insertion of the repeaters each satisfy a desired clock frequency (Ex-1005, 24), a POSA would understand that the clock rate for the entire chip is not controlled by longest and slowest wire (Ex-1005, 23). As discussed above, a chip operating at a higher clock rate would generally be able to perform more operations over a given time period than the same chip operating at a slower clock rate.

219. Hassoun teaches that its registers store data along the path, with each register having a delay of a clock period. Ex-1011, 249 (“The *realizable delay between consecutive registers on a path will always be determined by the clock period*, regardless of the actual signal propagation time. Register-to-register sub-paths with *delays larger than the permissible clock cycle are not permitted*.”). Thus, the repeaters/registers taught by Hassoun are *data storage elements*. Ex-

1011, 249; *see also* Ex-1005, 24-25; Ex-1010, 1. Each repeater causes a delay of one clock cycle on the channel. Ex-1011, 249; *see also* Ex-1005, 24-26; Ex-1010, 2, 3;

220. Hassoun teaches inserting three repeaters into the communication channel, where the data transfer delay has exceeded a predefined delay threshold (e.g., when “multiple clock cycles are required”). Ex-1011, 248-249. This results in introducing a delay of 3 cycles on the communication channel. Ex-1011, 249 (“The s-t path has *three registers* between s and t. It thus *takes four clock cycles to traverse* from s to t despite the different spacing between consecutive registers.”).

221. A POSA would have been motivated, with a reasonable expectation of success, to modify Dielissen in view of the teachings of Hassoun to first identify a data transfer delay exceeding a predefined threshold (given the known long wire problem) and to then predictably improve the chip by adding repeaters as taught by Hassoun to allow or maintain a higher clock rate within the chip. Modifying Dielissen to add the three repeaters as taught by Hassoun was an obvious way to improve Dielissen. Another paper by the authors of Dielissen regarding the Aethereal NoC expressly suggests adding repeaters to the Aethereal NoC to solve the long wire problem. Ex-1012, 3 (“*long wires can be pipelined*”). Further, it would have been obvious to a POSA that a large SoC such as that described by Dielissen would need to have links with multiple repeaters. Ex-1011, 1; Ex-1010,

1. Hassoun teaches that “[m]ultiple clock cycles will be required to cross a chip.” Ex-1011, 1. In some chips only “between 0.4 and 1.4%” of the chip area is reachable in one clock cycle.” Ex-1010, 1. A POSA would thus understand that links crossing between components on different sides of a chip would require multiple repeaters. Ex-1011, 1; Ex-1010, 1.

222. A POSA would also have been motivated to modify Dielissen in view of Hassoun to implement the specific number of data storage elements in a communication channel that corresponds to an integer multiple of the number of data elements, because some networks with fixed-length packets required “synchronization of packet transmissions.” Ex-1014, 6:53-67 and 12:21-27.

Dielissen teaches that the Aethereal NoC was such as network. Ex-1007, 1 (“This requires a logical notion of synchronicity, where all routers and NIs are in the same slot.”). This is confirmed in Rijpkema, another paper regarding Aethereal. Ex-1012, 1 (“communication links of a NoC are relatively shorter than those in computer networks, allowing tight synchronization between routers.”), 3 (“This requires a logical notion of router synchronicity, which is possible because a NoC is better controllable than a general network.”), 3 (“There is a logical notion of synchronicity; all routers in the network are in the same fixed-duration slot.”).

223. The ’2893 patent also admits that such systems exist. Ex-1001, 1:54-57 (“Such a network operates on the principle that a network station such as a data

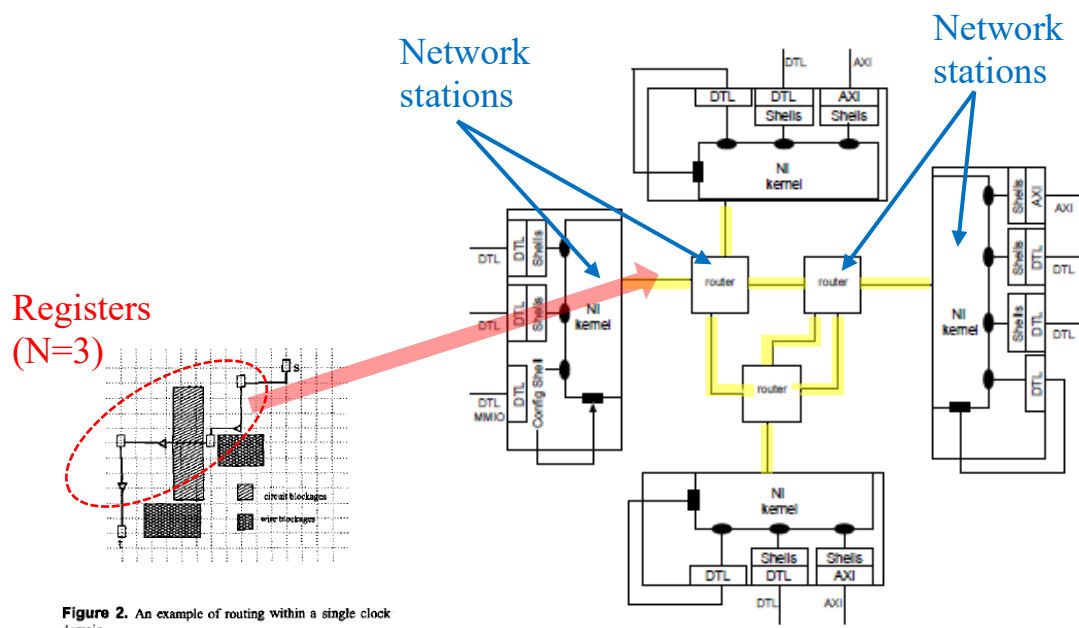
router receives the routing data incorporated in the data packet during a predefined clock cycle.”). In such system requiring synchronization of packet transmission, it was known to adjust the “transmission delay between adjacent nodes to be equal to an integral number of the packet transmission time” allowing each node to know “when to expect the header” of the packet. Ex-1014, 6:57-67. A POSA would understand that the transmission time for a flit as disclosed by Dielissen would be the number of words in the flit, i.e., $N=3$ cycles. Ex-1007, 2.

224. Adding an integer multiple of 3 cycles of delay would thus require an integer multiple of 3 repeaters placed into the communication channel (i.e., 3 repeaters, or 6 repeaters, or 9 repeaters, etc.). Adding such a delay would allow the router to know “when to expect a header.” Ex-1014, 6:57-67. Thus, for this reason also claim elements [10.e] and [10.f] would have been obvious under both Petitioner Patent Owner’s construction (requiring a data package of a fixed length of $N=3$ data elements) and Patent Owner’s construction (requiring only a data package *having at least* $N=2$ data elements).

225. As I stated above regarding element [10.pre.c], Dielissen teaches a fixed size data package of $N=3$ data elements. *Supra*, [10.pre.c]. M is described by the patent (and the claim) as M is any positive integer. For example, see Figs. 1 and 3 of the ‘2893 patent, and Ex-1001 at 3:50-51, 5:63-6:127:11-14. So to a POSA, the straightforward combination of Dielisson’s integrated circuit designed for data

packages with three data elements with Hassoun's teaching of identifying a data transfer delay exceeding a predetermined delay threshold and then inserting three data storage elements in a communication link (as explained above), with an M of one, results in the elements of [10.e] and [10.f]. That is because in view of the teachings a POSA would be "*introducing $M*N$ data storage elements*" ($1*3=3$ data storage elements) into the data communications network, for "*introducing a delay of $M*N$ cycles*" ($1*3=3$ delay cycles) on the communication channel.¹⁷ The below diagram shows an example of inserting three data storage elements as taught by Hassoun on one communication channel of Dielissen, but it would be equally apparent to a POSA to insert them on any other of the communication channels.

¹⁷ I note that Dielissen also teaches subchannels for communicating data in both directions between the network stations. *Infra*, §[7.g]. To a POSA, it would have been obvious to implement Hassoun's repeaters in both subchannels. In that instance, $M=2$ and $N=3$.



Ex-1007, Fig. 1; Ex-1011, Fig. 2. This combination would result in Dielissen designed to have three data storage elements, as Hassoun teaches, in a communication channel between its routers, which is $M \times N$ data storage elements as claimed ($M=1$ and $N=3$). See Ex-1007, Fig. 1; Ex-1011, Fig. 2 and p.3. A POSA would have expected this known option to predictably improve Dielissen.

226. A POSA would have had a reasonable expectation of success in inserting repeaters as taught by Hassoun because doing so was no more than using a known solution for a known issue. The references I discussed above, including Hassoun, Dall’Osso, and Bertozzi (Exhibits 1011, 1010, and 1005) show this. Also, Hassoun is in the same field of on-chip technology as Dielissen. Ex-1011, 1 (discussing “SoC design”); Ex-1007, 1 (discussing “SoC communication infrastructures, such as the Aethereal network on chip (NoC)”). Further, pipelining

(using repeaters) to improve the long wire / global wire delay problem was already known specifically in NoC applications. Ex-1005, 1, 3, 7-8; Ex-1010, 1-3. In fact, another paper by the authors of Dielissen also regarding the Aethereal NoC discloses that pipelining was an option on long links. Ex-1012, 3 (“Long wires can be pipelined”). In my opinion, adding M*N repeaters as claimed to the long wires in the Aethereal NoC to improve the long wire delay problem would have been merely the application of a known solution using only routine skill in the art at the time, to predictably improve Dielissen.

227. Under Patent Owner’s broader construction, inserting the data storage elements as recited by elements [10.e] and [10.f] would also have been obvious for a POSA to try. As I discussed above, the long wire problem was a known issue with a finite number of predictable, known options for solving it—inserting a small number of repeaters (“data storage elements”), depending on wire length.

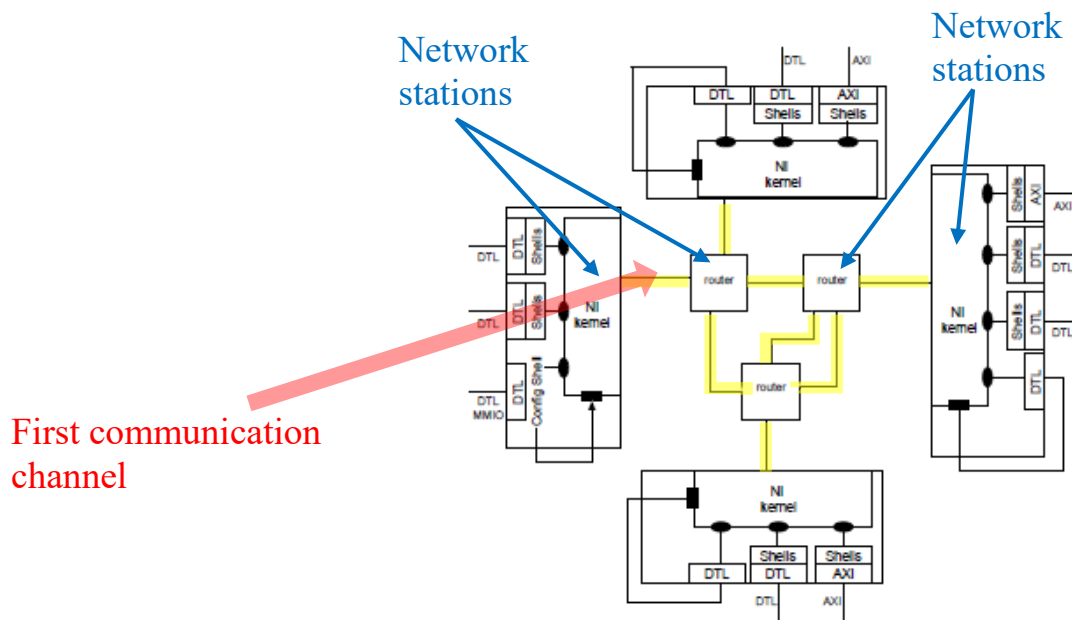
Examples of this include: Ex-1011, 1, 3; Ex-1010, 1-3, Fig. 2; Ex-1005, 3, 5-6, 7-8. A POSA would know, based on these references and general knowledge in the art as of early 2005, that the predictable set of options included one repeater, or two repeaters, or three repeaters, or four repeaters. See Ex-1011, 1, 3; Ex-1010, 1-3, Fig. 2; Ex-1005, 3, 5-6, 7-8. Each of these options would have been obvious to a POSA. Under Patent Owner’s construction, three of the four of those options, including inserting two, three, or four repeaters, would meet the M*N requirements

of element [10.f]. Two repeaters would meet the limitation under Patent Owner's construction requiring data packages having at least $N=2$ data elements where $N=2$ and $M=1$. Similarly, four repeaters would meet the limitation under Patent Owner's construction requiring data packages having at least $N=2$ data elements where $N=2$ and $M=2$. Finally, three repeaters would meet the limitation under both Petitioner Qualcomm's and Patent Owner's constructions where data packages have a size of $N=3$ where $N=3$ and $M=1$. Therefore, claim 10 would have been obvious for this additional reason.

228. [1.e]/[7.e]

Claim language	Evidence and disclosure
[1.e]/[7.e] the data communication network comprising a first network station and a second network station interconnected through a first communication channel	This element is disclosed for the same reasons as discussed above regarding [10.e] and [10.f] under both Petitioner Qualcomm's and Patent Owner's construction for the same reasons I discussed above in [10.e] and [10.f].

229. I have annotated an exemplary first communication channel between two network stations on Figure 1 of Dielissen below. My selection of the communication channel is only exemplary, and other channels could have been selected as well.



Ex-1007, Fig. 1 (red, blue, and yellow annotations added).

230. [7.f]

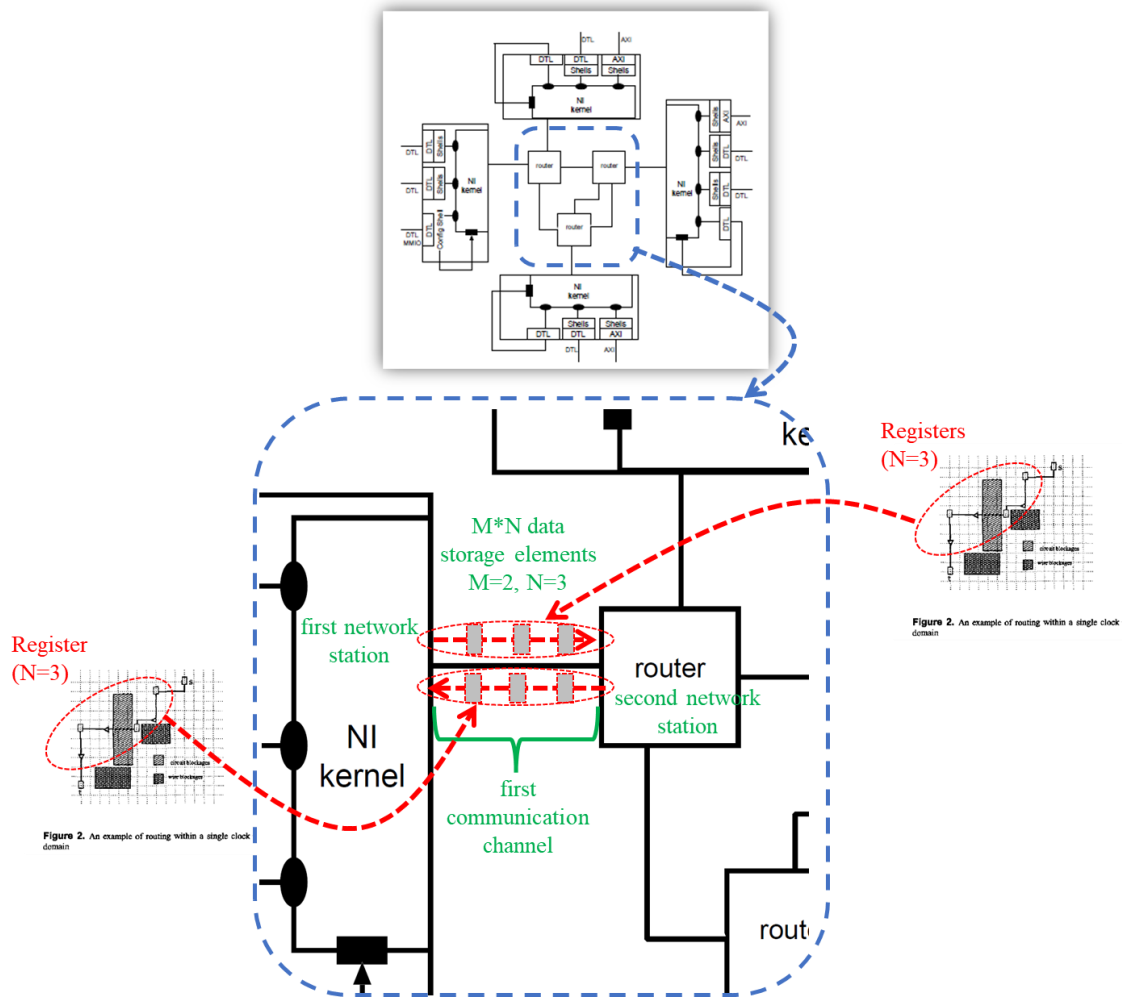
Claim language	Evidence and disclosure
[7.f] the data communication network further comprising $M \times N$ data storage elements, M being a positive integer, for introducing a delay of $M \times N$ cycles on the first communication channel,	Dielissen does not disclose this element. However, Hassoun could be combined with Dielissen in order to teach this element under both Petitioner Qualcomm's and Patent Owner's construction for the same reasons I discussed above in [10.e] and [10.f].

231. Dielissen does not disclose this element. However, Hassoun could be combined with Dielissen in order to teach this element under both Petitioner

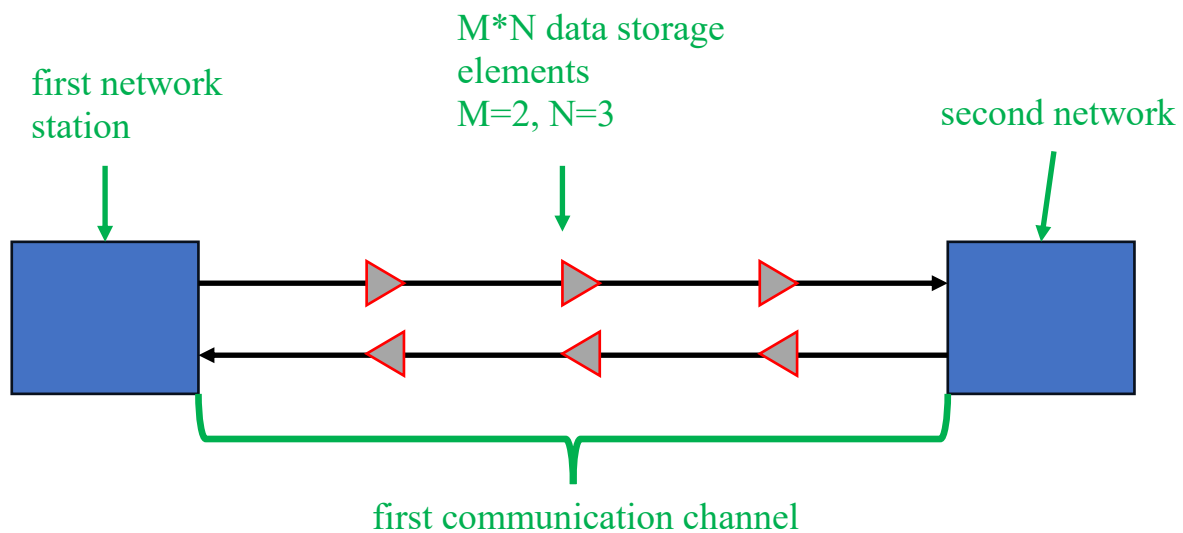
Qualcomm's and Patent Owner's construction for the same reasons I discussed above in [10.e] and [10.f].

232. Under Petitioner Qualcomm's and Patent Owner's construction the combination of Dielissen with Hassoun would result in the NoC ("data communications network") further comprising 6 repeaters ("data storage elements" where $M=2$, $N=3$)¹⁸ for introducing a delay of 6 cycle ($M*N$ cycles) on the first communication channel. *Supra*, [10.e] and [10.f]; (see also [7.g] below showing a POSA's understanding of Dielissen disclosing bidirectional communication). I have illustrated this below:

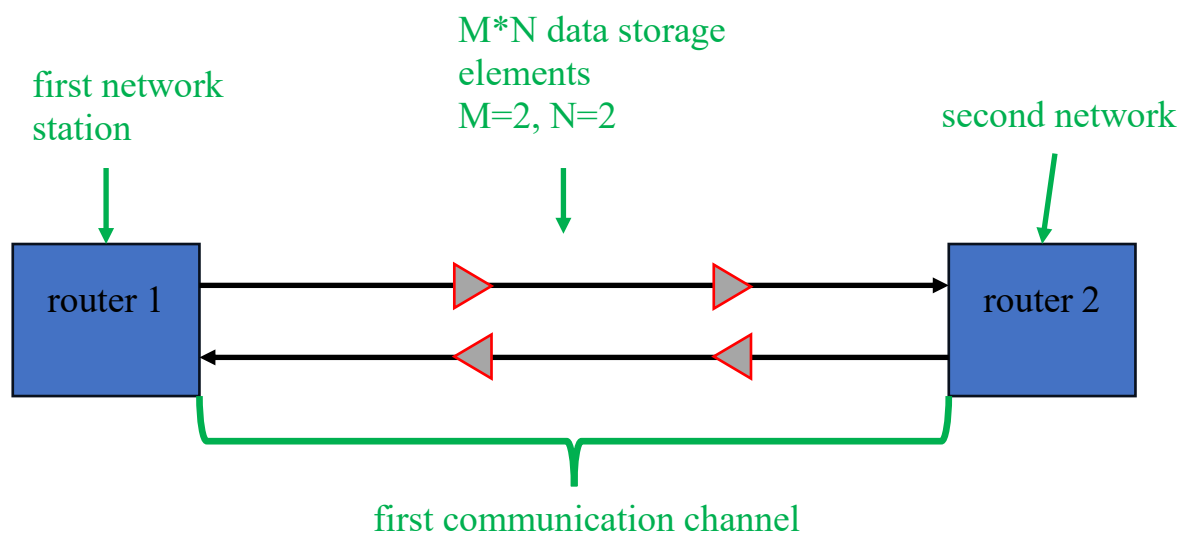
¹⁸ $M=2$ being a positive integer.



Ex-1007, Fig. 1; Ex-1011, Fig. 2 (annotated). Which I simplified as:



233. Alternatively, under Patent Owner's construction the combination of Dielissen with Hassoun would result in the NoC ("data communications network") further comprising 4 repeaters ("data storage elements" where $M=2$, $N=2$)¹⁹ for introducing a delay of 4 cycle ($M*N$ cycles) on the first communication channel. *Supra*, [10.e] and [10.f]. Similar to the illustration shown above, I have illustrated this below but with $M=2$ and $N=2$:



¹⁹ $M=2$ being a positive integer. I disagree with Patent Owner's construction of "at least N data elements" at least because it would allow for this unbounded ability to interpret what a data package is, as illustrated in the example herein where simply setting $N=2$, an integer, would result in such a nonsensical example since the '2893 specification is clear that data packages have uniform, equal size (fixed size) as I explained previously.

234. [1.f]

Claim language	Evidence and disclosure
[1.f] the data communication network further comprising M*N data storage elements, M being a positive integer, the data communication introducing a delay of M*N cycles on the first communication channel when the data communication network identifies the first communication channel as having a data transfer delay exceeding a predefined delay threshold.	I have been instructed to apply the same scope to element [1.f] as elements [10.e] and [10.f], and I have done so. Therefore, element [1.f] is met by Bertozzi for the same reasons that I discussed above with respect to [10.e] and [10.f].

235. [7.g]

Claim language	Evidence and disclosure
[7.g] wherein each communication channel of the plurality of communication channels comprises: a first subchannel between an input port of a first associated network station and	Dielissen discloses this element. For example: Ex-1007, 3 (“To avoid overflow in the BE-input queues, <i>a link-level flow control scheme is implemented.</i> Each router is initialized with the amount of free space in the connected routers and network interfaces. <i>Every time a flit is send to a next router, the free space counter corresponding to that destination port is decremented.</i> When a router schedules a flit for the next slot, <i>it signals its predecessor that the free space counter can be</i>

an output port of a second associated network station; and a second subchannel between an input port of the second associated network station and an output port of the first associated network station, and	incremented. Since GT-packets follow a pipelined circuit, a GT-flit is always send to the next router in the next cycle, and therefore link-level flow control can be omitted.”) • Ex-1007, 3 (“The NI kernel communicates with the NI shells via ports. At each port, <i>peer-to-peer connections can be configured</i>, their number being selected at NI instantiation time. A port can have multiple connection to allow differentiated traffic classes (e.g., best effort, or guaranteed throughput), in which case there are also connid signals to select on which connection a message is supplied or consumed.”) • Ex-1007, 3 (“For each connection, <i>there are two message queues (one source queue, for messages going to the network, and one destination queue, for messages coming from the network)</i> in the NI kernel. Their size is also selected at the NI instantiation time. Queues provide the clock domain crossing between the network and the IP modules. Each port can, therefore, have a different frequency”) • Ex-1007, 3 (“End-to-end flow control ensures that no data is sent unless there is enough space in the destination buffer to accommodate it. This is implemented using credits [17]. For each channel, there is a counter (Space) tracking the empty buffer space of the remote destination queue. This counter is configured with the remote buffer size. When data is sent from the source queue, the counter is decremented. When data is consumed by the IP module at the other side, credits are produced in a counter (Credit) to indicate that more empty space is available. These credits are sent to the producer of data to be added to its Space counter. <i>In the Aethereal prototype, we piggyback credits in the header of the packets for the data in the other direction to improve network efficiency.</i> Note that at most Space data items can be transmitted. We call sendable data,
--	--

	the minimum between the data items in the queue and the value in the counter Space.”) • Ex-1007, 5 (“In Figure 12, we show the necessary steps in setting up a connection between two modules (master B and slave A) from a configuration module (Cfg). Like for any other memory-mapped register, before sending configuration messages for programming the B to A connection, a connection to the remote NI must be set up. <i>This involves two channels, one for the requests and and [sic] one for the responses between NI1 and the configuration port of NI2.</i> This connection is opened in two steps. First, the channel to the remote NI configuration port is set up by writing the necessary registers in NI1 (Step 1 in Figures 11 and 12). Second, we use this channel to set up (via the network) the channel from configuration port of NI2 to the configuration port of NI1 (Step 2). The three shown messages are delivered and executed in order at NI2. The last of them also requests an acknowledgment message to confirm that the channel has been successfully set up.”) • Ex-1007, Figs. 6, 12
--	--

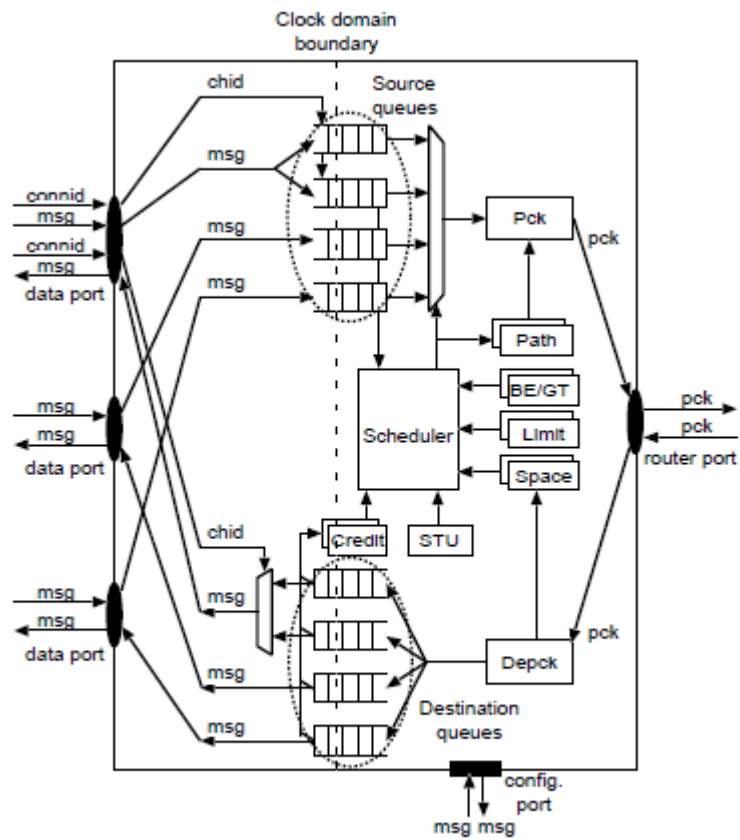


Figure 6. Network interface kernel

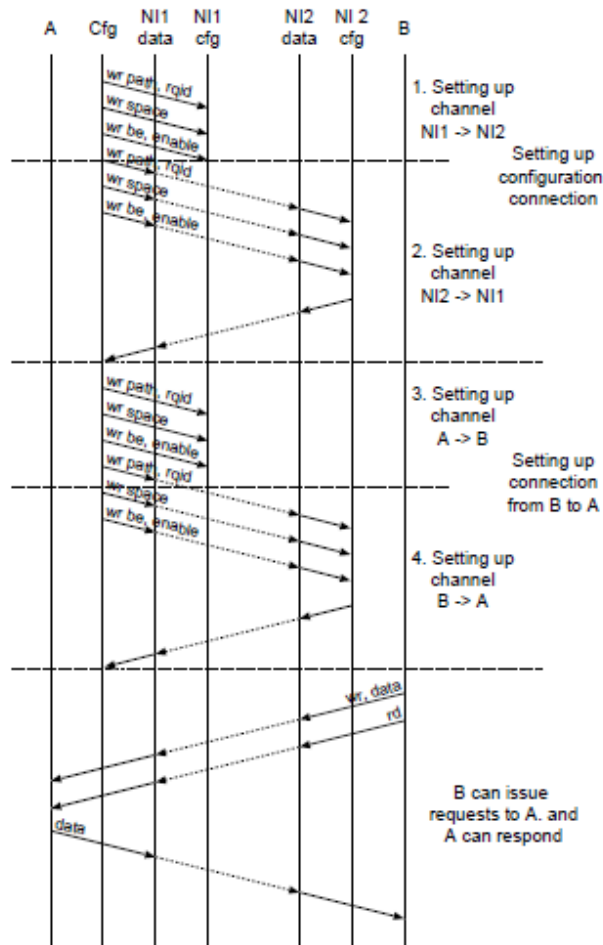


Figure 12. Connection configuration example

236. Each link (“communication channel”) disclosed by Dielissen is bidirectional, meaning that it would have subchannels for communicating data in each direction between the network stations. Ex-1007, 3, Fig. 6. First, Figure 6 shows a network interface with data ports (for communicating with various functional blocks) and a router port. Ex-1007, 3, Fig. 6. The router ports show both input and output ports, implying that both the network interface and routers have

bidirectional communication with both input and output ports. Ex-1007, Fig. 6. I have annotated Figure 6 below to show this:

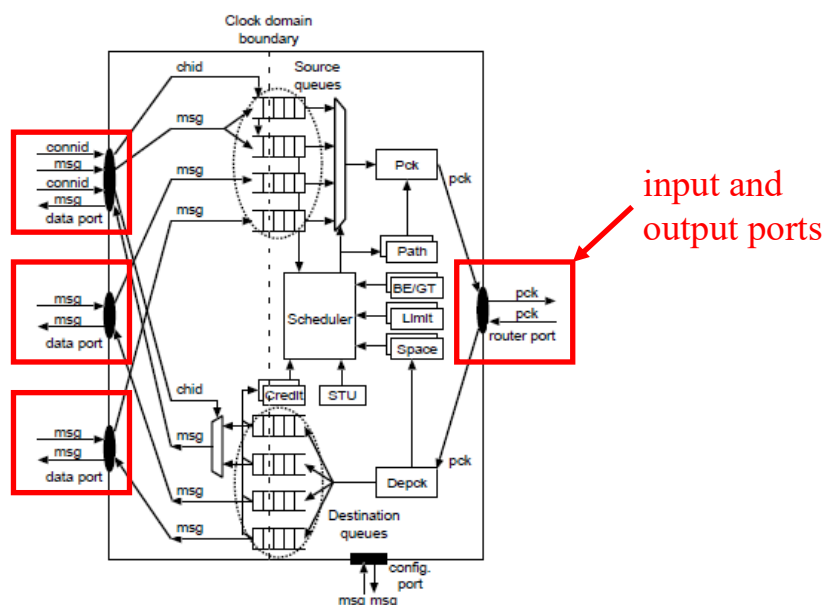


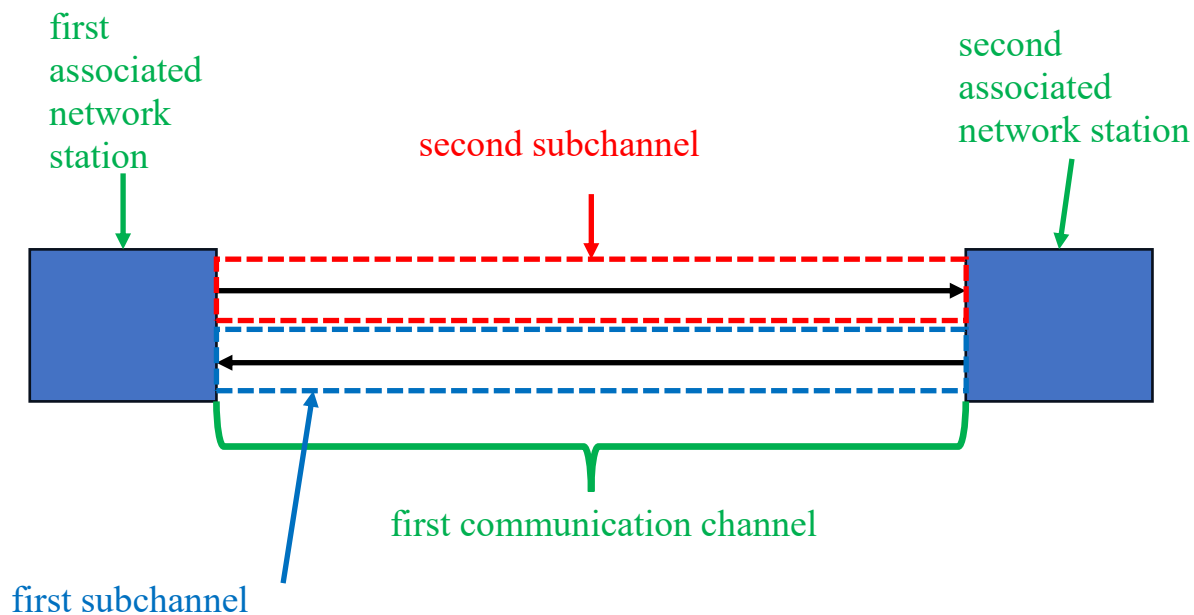
Figure 6. Network interface kernel

Ex-1007, Fig. 6 (red annotations added). Second, Dielissen teaches that end-to-end connections have both a request and a response channel. Ex-1007, 5 (“This involves two channels, one for the requests and one for the responses between NI1 and configuration port of NI2”). A POSA would understand the individual sublinks used within the end-to-end connections would also have a request and response link. Third, Figure 12 of Dielissen shows the “path” only transmitted from NI1 side to NI2 side. Ex-1007, Fig. 12. A POSITA would understand that the messages from NI2 to NI1 would necessarily traverse the same path back indicating that every link is bidirectional. Fourth, Dielissen’s NoC uses link-level flow control. Ex-1007, 1, 3. Link-level flow control requires connected network stations to

communicate regarding free space within their buffer. Ex-1007, 3 (“Every time a flit is send to a next router, the free space counter corresponding to that destination port is decremented. *When a router schedules a flit for the next slot, it signals its predecessor that the free space counter can be incremented.*”). A POSA would understand that this requires bidirectional communication at the link-level.

237. Such communication would require two subchannels: a first subchannel between the output port of the first network station to the input port of the second network station and a second subchannel between the input port of the second network station to the input port of the first network station. A POSA would understand that a sublink taught by Dielissen is connected between the output port of one network station and the input port of another network station as the routers and network interfaces taught by Dielissen have input and output ports. Ex-1007, 2-3, Figs. 2, 6, 7.

238. I have illustrated an example link below:

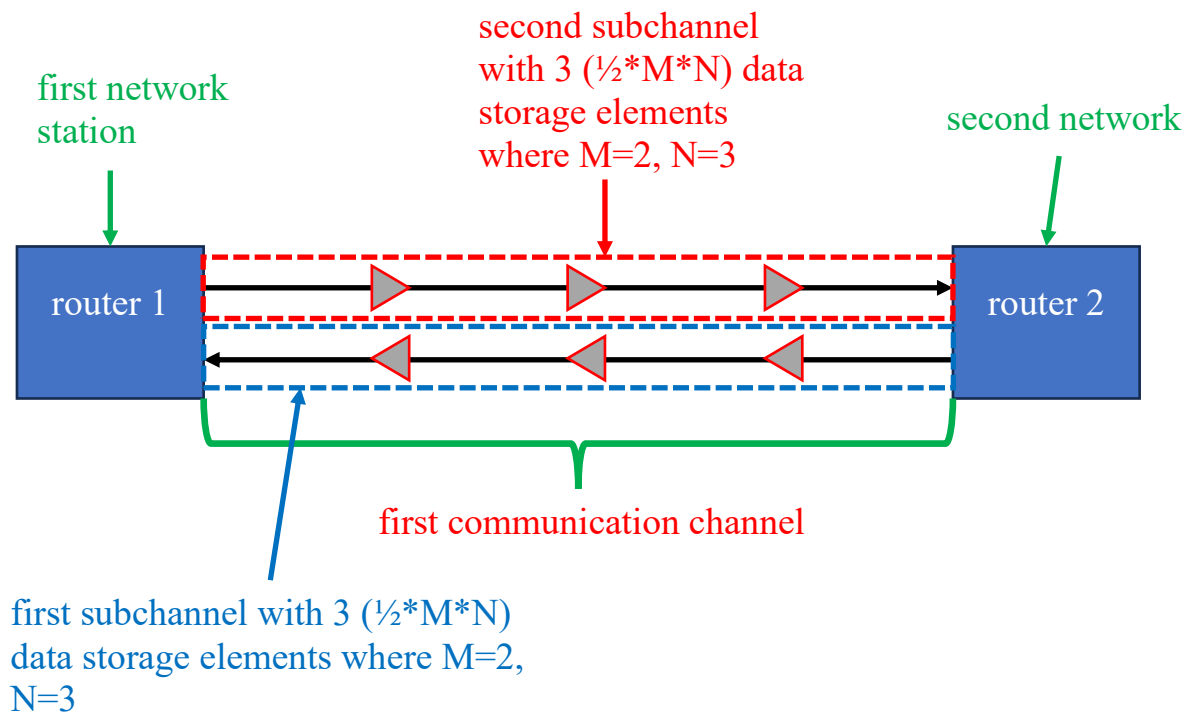


239. [7.h]

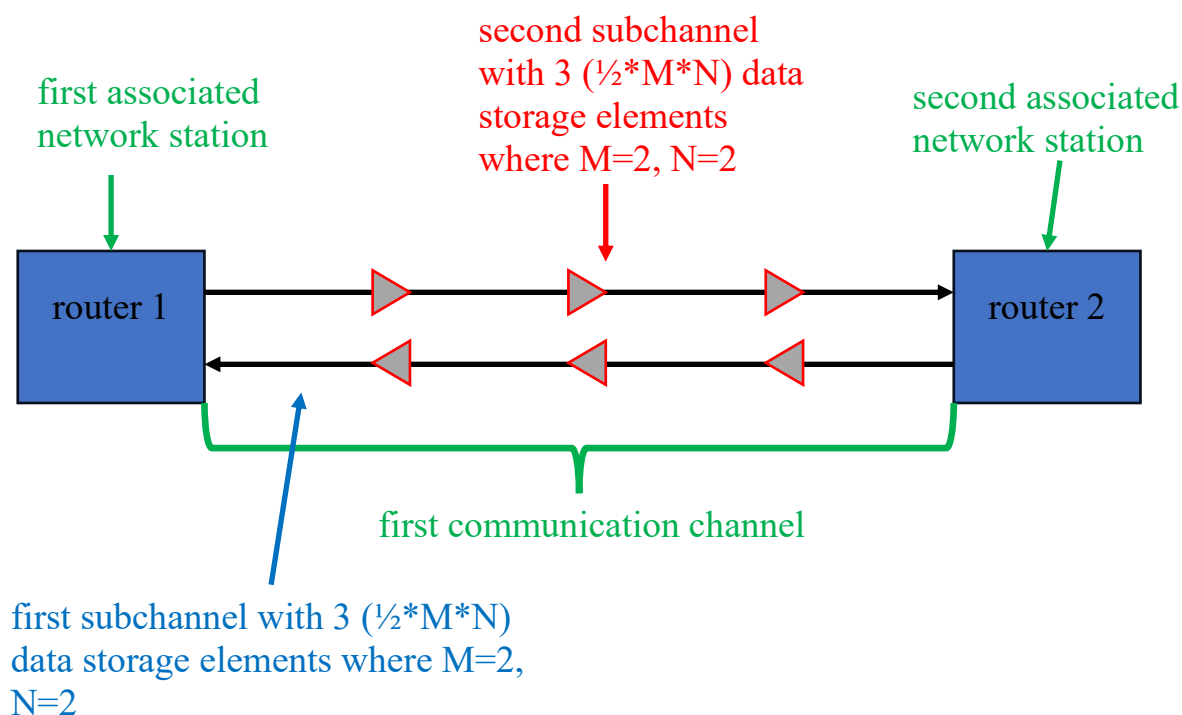
Claim language	Evidence and disclosure
[7.h] wherein the first subchannel and the second subchannel of the first communication channel each comprise a delay introduced by $\frac{1}{2} * M * N$ data storage elements	This element is disclosed by the combination of Dielissen with Hassoun for the reasons discussed in [7.f].

240. Under Petitioner Qualcomm’s construction and Patent Owner’s construction, the obvious combination of Dielissen and Hassoun discloses this element where $M=2$ and $N=3$. Both the first subchannel and the second subchannel

of the first link (“communication channel”) comprise a delay introduced by 3 ($\frac{1}{2} * M * N$) repeaters (“data storage elements”). I have illustrated this below:



241. Under Patent Owner’s construction, this element would be met for the same reasons as I discussed above regarding Petitioner Qualcomm’s construction. Alternatively, the obvious combination of Dielissen and Hassoun discloses this element where $M=2$ and $N=2$. Both the first subchannel and the second subchannel of the first link (“communication channel”) comprise a delay introduced by 2 ($\frac{1}{2} * M * N$) repeaters (“data storage elements”). This is illustrated below:



242. Dependent Claim 2:

Claim language	Evidence and disclosure
[2] The integrated circuit as claimed in claim 1, wherein each communication channel of the plurality of communication channels comprises: a first subchannel between an input port of a first associated network station and an output port of a second associated network station; and a second subchannel between an input port	A POSTIA would understand that bidirectional communication would enable duplex communication between two associated network stations. The obvious combination of Dielissen and Hassoun teaches dependent claim 2 for the reasons explained above with respect to elements [7.g].

of the second associated network station and an output port of the first associated network station for enabling duplex communication between the two associated network stations.	
--	--

243. Dependent Claim 3:

Claim language	Evidence and disclosure
[3] The integrated circuit as claimed in claim 2, wherein the M*N data storage elements are distributed over the first subchannel and the second subchannel of the first communication channel.	The obvious combination of Dielissen and Hassoun teaches dependent claim 3 for the reasons explained above with respect to element [7.h].

244. Dependent Claim 6:

Claim language	Evidence and disclosure
[6] The integrated circuit as claimed in claim 1, wherein M=1.	The obvious combination of Dielissen and Hassoun teaches dependent claim 6 for the reasons explained above with respect to elements [10.e] and [10.f].

245. Dependent Claim 9

Claim language	Evidence and disclosure
[9] The integrated circuit as claimed in claim 1, wherein each network station comprises a first memory element and a second memory element associated with a communication port of said network station, the first memory element being arranged to store routing information from a data package requiring a first service level associated with said communication port and the second memory element being arranged to store routing information from a data package requiring a second service level associated with said communication port.	This is disclosed by Dielissen. All elements of claim 1 are disclosed as discussed above. Dielissen discloses that each network station (“routers and network interfaces”) comprises a first queue (“memory element”) and a second queue (“memory element”) associated with a communication port of said network station, the first queue (“memory element”) being arranged to store routing information a data package requiring a guaranteed throughput service (“first service level”) associated with said communication port and the second queue (“memory element”) being arranged to store routing information from a data package requiring a best effort service (“second service level”) associated with said communication port. For example: Ex-1007, 1-2 (“The <i>Æthereal</i> NoC provides both <i>best-effort and guaranteed services (e.g., latency or throughput)</i>. To <i>implement guarantees</i>, we use contention-free routing, which is based on a time-division multiplexed circuit-switching approach, where one or more circuits are set up for a connection [14]. This requires a logical notion of synchronicity, where <i>all routers and NIs</i> are in the same slot. Circuits are created by <i>reserving consecutive slots in consecutive routers/NIs</i>. This is, the circuits are pipelined, in the sense that if a circuit is set from router R to router R’, and slot s is reserved at router R, then slot s + 1 must be reserved at router R’. On these circuits, data received in one slot will be forwarded to the next router/NI in the next slot. By <i>setting up circuits</i>, we ensure that data is transported without contention. In this way throughput and latency are guaranteed. We call this guaranteed traffic as

	<i>guaranteed throughput (GT)</i> data, as opposed to the <i>best-effort (BE)</i> data, for which no throughput guarantees are given.”) - Ex-1007, 2 (“As mentioned above, <i>circuits are set up by reserving slots</i>. These slots are reserved such that <i>no more than one GT</i> data is scheduled at the same time on an <i>output port of a router or NI</i>. <i>BE data is transferred on the slots that are not used by the GT data</i>: either the slots are unreserved, or the slots are reserved, but not used. <i>BE data can be delayed because of the higher priority GT data, or because of contention on the ports.</i>”) Referring to Figure 2: - Ex-1007, 2 (“We select <i>source routing as an addressing scheme</i>, because it allows topology independence, while at the same time has a low cost: no expensive (programmable) lookup tables are needed in the router. In <i>source routing the path on which the packet travels is included in the header of a packet</i>. In <i>Æthereal</i>, <i>this path is a list of destination ports</i>, from which <i>each router on the path removes the first element for its own use.</i>”) - Ex-1007, 2 (“Besides the path, the header also contains information for the Network interface, which is explained in section 2.2.1.”) - Ex-1007, 2 (“<i>GT and BE-flits are semantically the same</i>, but they are handled differently by the scheduler: GT-flits are always scheduled for the next cycle. The BE-flits are scheduled to the remaining destination ports according a round-robin schedule. Ones[sic] a first flit of a BE-packet is send to a certain destination port, than port remains locked until the packet is finished: the port does not schedule BE-flits from the other input ports. In this way the interleaving of BE-packets is avoided, which makes implementation simple and cheap. Note that BE-packets can still
--	---

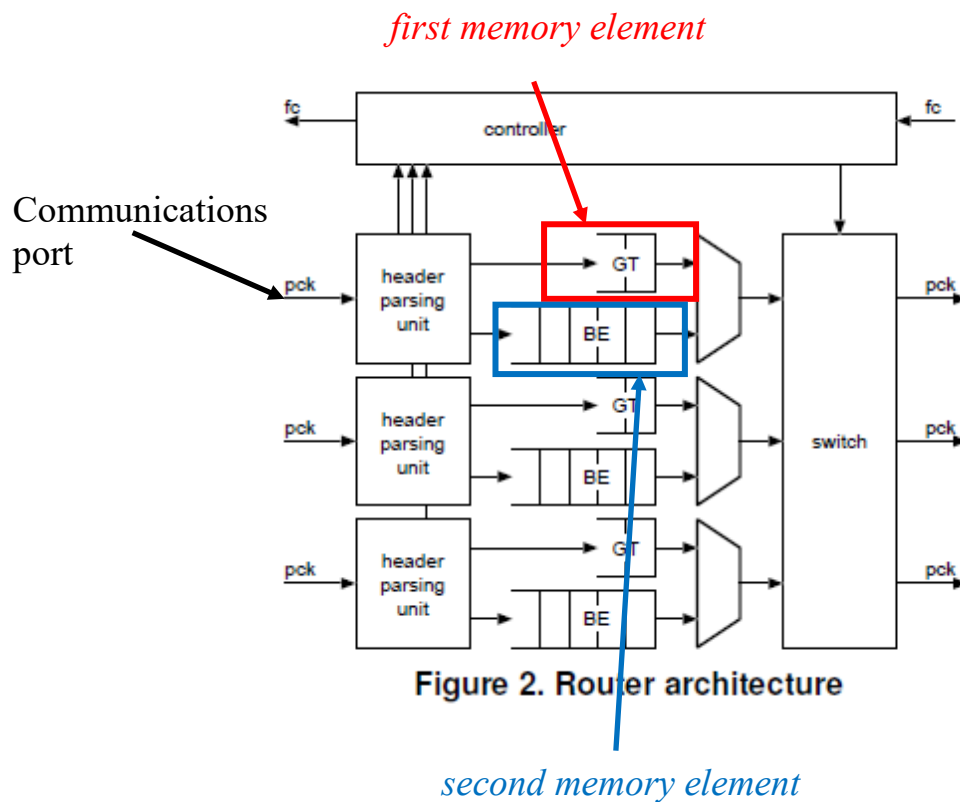
	be interleaved with GTpackets. For GT-flits the interleaving amongst themselves has to be avoided in the static schedule.”) With reference to the NI’s and particularly the NI Kernal (Section 2.2.1): • Ex-1007, 3 (“The <i>NI kernel</i> communicates with the NI shells via ports. At <i>each port</i>, peer-to-peer connections can be configured, their number being selected at NI instantiation time. <i>A port can have multiple connection to allow differentiated traffic classes (e.g., best effort, or guaranteed throughput)</i>, in which case there are also <i>connid signals to select on which connection a message is supplied or consumed.</i>”) • Ex-1007, 3-4 (“Each channel is configured individually. In a first prototype of <i>the Aethereal network interface</i>, we <i>can configure if a channel is best effort (BE) or providing timing guarantees (GT)</i>, reserve slots in the latter case, configure the end-to-end flow control, and the routing information.”) • Ex-1007, 4 (“From <i>the source queues</i>, data is packetized (Pck) and sent to the network via a single link. A <i>packet header consists of the routing information (NI address for destination routing, and path for source routing)</i>, remote queue id (i.e., the queue of the remote network interface in which the data will be stored), and piggybacked credits (see Figure 3).”) • Ex-1007, 4 (“There are <i>multiple channels</i> which may require data transmission, <i>we implement a scheduler to arbitrate between them.</i> A <i>queue becomes eligible for scheduling when either there is sendable data</i> (i.e., there is data to be sent, and is space in the channel’s destination buffer), or when there are credits to send. In this way, when there is no sendable data, it is still possible to send credits in an empty packet.”)
--	---

	• Ex-1007, 4 (“The <i>scheduler checks if the current slot is reserved for a GT channel</i>. If the slot is reserved and the GT channel is eligible for scheduling, then the channel is granted data transmission. Otherwise, the <i>scheduler selects an eligible BE channel using some arbitration scheme</i>: e.g. round-robin, weighted round-robin, or based on the queue filling.”) • Ex-1007, 4 (“Once a <i>queue is selected</i>, a packet containing the largest possible amount of credits and data will be produced. The amount of credits is bound by implementation to the given number of bits in the packet header, and packets have a maximum length to avoid links being used exclusively by a packet/channel, leading to congestion.”) • Ex-1007, 4 (“On the outgoing path, packets are depacketized, credits are added to the counter Space, and data is stored in its corresponding queue, which is given by a queue id field in the header.”) The '2893 patent discloses “best effort and guaranteed bandwidth” as two service levels that are an embodiment of claim 9: • Ex-1001, 3:23-33 (“Each network station may comprise a first memory element and a second memory element associated with a communication port of said network station, the first memory element being arranged to store routing information from a data package requiring a first service level associated with said communication port and the second memory element being arranged to store routing information from a data package requiring a second service level associated with said communication port. <i>Such an implementation allows for the interleaving of packages requiring different service levels, e.g.</i>
--	--

	<i>best effort and guaranteed bandwidth service levels.”)</i> Similar to the ’2893 patent, Dielessen discloses networks that support “guaranteed throughput (GT)” and “best-effort (BE)” service levels. Ex-1007, 1. Furthermore, each of the routers and NIs (i.e. each network station) has GT and BE queues (also referred to as channels in the NIs) with routing information.²⁰ Dielissen teaches the concept of GT and BE on all routers and network interfaces at least because Dielissen discloses the implementation of GT on “<i>all routers and NIs</i>” and “<i>BE data is transferred on the slots that are not used by the GT data.</i>” Ex-1007, 1-2.
--	---

246. Figure 2 is an example of a router architecture with first and second memories, which Dielissen refers to as queues, at least one for guaranteed throughput (GT) and a second for best effort (BE). Ex-1007, Fig. 2. Such queues store the data packages, including the data elements that contain the routing information. The memories contain the routing information because a router only “removes the first element for its own use.” Ex-1007, 2. I have annotated an exemplary set of memories below:

²⁰ I have discussed routing information in element [10.pre.c].



Ex-1007, Fig. 2 (annotated).

247. Similarly, Dielissen explains that the network interfaces have separate connections for guaranteed throughput and best effort traffic, and that each of these connections has a separate source queue.

A port can have multiple connection to *allow differentiated traffic classes* (e.g., *best effort, or guaranteed throughput*).... For each connection, there are *two message queues* (one source queue, for messages going to the network, and one destination queue, for messages coming from the network) in the NI kernel.

Ex-1007, 3. A POSA would therefore understand that Figure 6 shows a port where a “msg” (data package) is supplied to and stored in one source queue (first memory element) if it is a msg requiring guaranteed throughput (first service level) and a different source queue (second memory element) if it is a best effort msg (second service level). A POSA would understand that Figure 6, shows a port where based on connid, “msg” is supplied to either a first memory or second memory (*see* “Source queues” at top and “**channels** which may require data transmission” quotes below) in order to “*allow differentiated traffic classes (e.g., best effort, or guaranteed throughput)*, in which case there are also *connid signals to select on which connection a message is supplied or consumed.*” Ex-1007, Figs. 6, 3.

Figure 6 shows a “Scheduler” where based on “BE/GT” that control the top multiplexer, such selection is performed from the appropriate channel “The *scheduler checks if the current slot is reserved for a GT channel.* If the slot is reserved and the GT channel is eligible for scheduling, then the channel is granted data transmission. Otherwise, the *scheduler selects an eligible BE channel using some arbitration scheme:* e.g. round-robin, weighted round-robin, or based on the queue filling.” Ex-1007, 4. I have annotated an exemplary set of memories below:

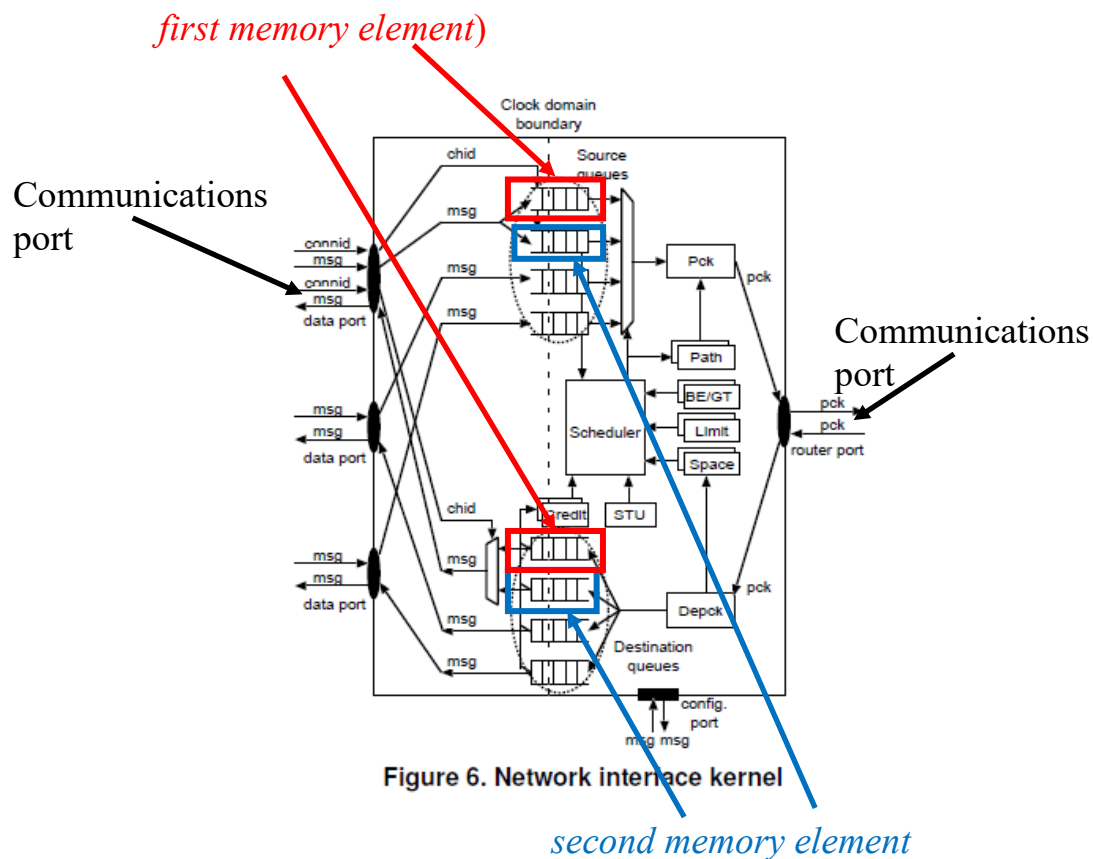


Figure 6. Network interface kernel

Ex-1007, Fig. 6 (annotated).

248. A POSA would understand that the queues are memories. Dielissen describes that these queues are implemented with FIFOs, which a POSA would have understood are a kind of memory. For example, Dielissen describes the size of a BE queue as “queues of 32-bit wide and 24-word deep each implemented using custom-made fifos.” Ex-1007, 6; *see also id.*, 3 (“32-bit wide and 24-word deep.”). The network interface queues are also described “being 32-bit wide and 8-word deep.” Ex-1007, 5; *see also id.*, 6 (“All queues are 32-bit wide and 8-word

deep, and are implemented using custom-made fifos.”). The “hardware fifos” are used instead of alternative types of memory. Ex-1007, 5 (“The queues are are-efficient custom made hardware fifos. We use these fifos instead of RAMs...”)²¹.

A POSA would understand that a fifo is a memory structure.

XIV. GROUND 4: GROUNDS 1-3 FURTHER IN VIEW OF SULLAM RENDERS CLAIM 8 OBVIOUS

249. It is my opinion that Grounds 1 through 3 disclose independent claim 1 for the reasons I discuss above. Claim 8 depends from claim 1.

250. Dependent Claim 8

Claim language	Evidence and disclosure
[8] The integrated circuit as claimed in claim 1, further comprising a clock signal generator, the plurality of network stations and the M*N data storage elements being responsive to the clock signal generator.	Grounds 1 through 3 further in view of Sullam disclose dependent claim 8. All other elements are taught for the reasons expressed in Grounds 1 through 3. Sullam discloses on-chip clocking circuitry. Ex-1013, 1:51-56 (“Specifically, the <i>microcontroller is typically defined as a single chip (integrated circuit)</i> that includes a processor, volatile memory such as Random Access Memory (RAM), non-volatile memory such as Read Only Memory (ROM), clocking source, and an input/output control unit. As such, <i>the microcontroller is a computer on a chip.</i>”), Ex-1013, 6:15-16 (“The microcontroller 100 of FIG. 1 also includes <i>clocking circuitry 112</i>”); Fig. 1. “[C]locking architecture 112 is able to <i>provide[] a wide variety of clocking signal frequencies to different circuitry that it is couple to.</i>” Ex-1013, 6:18-21. The clocking architecture includes an on-chip oscillator. Ex-1013, 7:35-37 (“The clocking

²¹ A POSA would understand that RAM means Random Access Memory.

architecture 112 also *includes the 24 MHz oscillator 212 which may be implemented with a precision integrated circuit (IC) oscillator.*”). A POSA would understand the clocking circuitry taught by Sullam to be a clock signal generator.

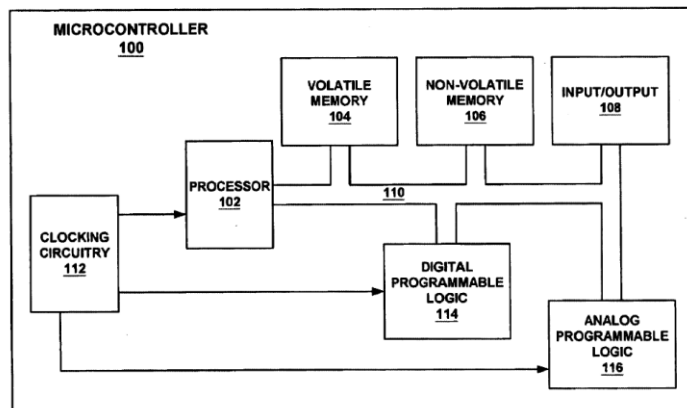


FIGURE 1

Ex-1013, Fig. 1.

Regarding Bertozzi (Grounds 1 and 2)

A POSA would understand that Bertozzi discloses a singular clock signal that all the switches/routers and network interfaces (including the pipeline stages) (“the plurality of network stations”) and repeaters (M*N data storage elements) are responsive to a singular clock.

For example:

- Ex-1005, 7 (“desired clock speed of the design”)
- Ex-1005, 7 (“latency of a channel connecting two modules may end up being more than *one clock cycle*”)
- Ex-1005, 7 (“*desired clock frequency that needs to be achieved in the design*”)
- Ex-1005, 7 (“a certain number of *clock cycles* is needed by a flit to cross the interconnect”)
- Ex-1005, 8 (“Multiple outstanding flits propagate across the link during the *same clock cycle*”)

	• Ex-1005, 10 (“As can be observed <i>from Fig. 6</i>, each <i>output module is deeply pipelined (7 pipeline stages)</i> so to maximize <i>the operating clock frequency of the switch.</i>”) • Also, a POSA would understand that Bertozzi also discloses referencing Fig. 9 “...nodes (with their <i>network interfaces</i>), <i>links</i> and <i>switches</i>” that have a singular “<i>clock speed target</i>,” for the entire design to be generated by <i>XpipesCompiler</i>. Ex-1005, 12. • Bertozzi, when describing Fig. 4 propagation of pipeline stages through the switch and delay stages in the link: “Fig. 4 illustrates the link model, which is <i>equivalent to a pipelined shift register</i>. Pipelining has been used both for data and control lines. Multiple outstanding flits <i>propagate across the link during the same clock cycle</i>. When flits are correctly received at the destination switch, an ACK is propagated back to the source, and <i>after N clock cycles (where N is the link length expressed in terms of number of repeater stages)</i> the flit will be discarded from the buffer of the source switch. On the contrary, a corrupted flit is NACKed and will be retransmitted in due time.” Ex-1005, 8. A POSA would understand this to require the repeaters and switches to operate under the same clock. • Furthermore, when describing the operation of the entire system, Bertozzi references uniform timing “cycles” to describe the switch and links. Referencing Fig. 5 switch, “For latency insensitive operation, the switch has virtual channel registers to store $2N + M$ flits, where <i>N is the link length (expressed as number of basic repeater stages)</i> and <i>M is a switch architecture related contribution (12 cycles in this design)</i>. The reason is that each transmitted flit has to be acknowledged before being discarded from the buffer. Before an ACK is received, the <i>flit has to</i>
--	---

	<i>propagate across the link (N cycles)</i>, an ACK/NACK decision has to be taken at the destination switch (<i>a portion of M cycles</i>), the ACK/NACK signal has to be <i>propagated back (N cycles)</i> and recognized by the source switch (<i>remaining portion of M cycles</i>). During this time, other $2N + M$ flits are transmitted but not yet ACKed.” Ex-1005, 8-9. <u>Regarding Ground 3</u> A POSA would have understood that the routers/switch and network interfaces (“plurality of network station”) of Dielissen and the repeaters (“M*N data storage elements”) of Hassoun would be responsive to various clock signals. Dielissen discusses ports having different frequencies and clock domains. It discloses that the NI also “implement[s]... <i>clock domain crossing</i>,” which implies multiple clock domains that may or may not be different clocks that can be disabled or enabled separately and/or be different frequencies. Ex-1007, 3. It further discloses the NI having “Queues provide <i>the clock domain crossing</i> between the <i>network and the IP modules</i>. Each <i>port can, therefore, have a different frequency</i>.” Ex-1007, 3. Dielissen goes on to say that in the network interface, the “fifos also implement the clock domain boundary allowing NI ports to run at a different frequency <i>than the network</i>.” Ex-1007, 6. These quotes imply that the network is in one clock domain. A POSA would thus understand that the network interfaces and routers of Dielissen, and the repeaters of Hassoun would be responsive to the same clock. Figure 6 shows a “Clock domain boundary” across the FIFOs (i.e. queues).
--	---

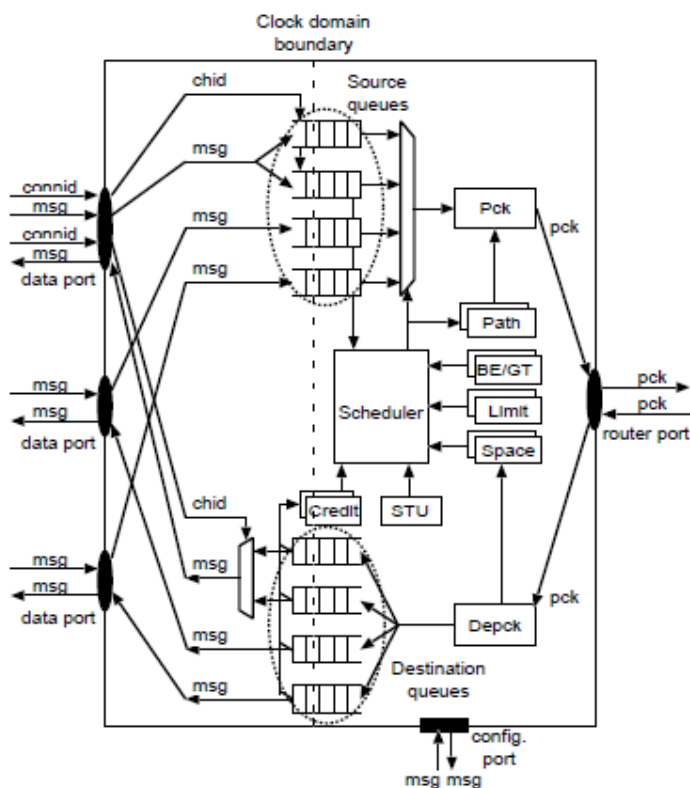


Figure 6. Network interface kernel

Dielessen also discusses the implementation: “[w]e have *synthesized an instance of a NI kernel* with a slot table of 16 slots, and 4 ports having 1, 1, 2, and 4 channels, respectively, with all queues being 32-bit wide and 8-word deep. The queues are area-efficient custom-made *hardware fifos*. We use these fifos instead of RAMs, because we need simultaneous access at all NI ports (possibly *running at different speeds*) as well as simultaneous read and write access for incoming and outgoing packets, which cannot be offered with a single RAM. Moreover, for the small queues needed in the NI, multiple RAMs have a too large area overhead. Furthermore the *hardware fifos implement the clock domain boundary* allowing *each NI port to run at a different frequency*. The rest of the NI kernel runs at a frequency of 500 MHz, and delivers a bandwidth towards the router of 16 Gbit/s in each direction. The

	synthesized area for this NI-kernel instance is 0.13 mm² in a 0.13μm technology.” Ex-1007, 5. A POSA would understand that an on-chip clock signal generator can generate the various clocks needed on an IC from a single sourced clock (or even generate the source clock on-chip via internal oscillators). For example, a single clock signal generator could produce multiple clocks of different frequencies. Ex-1013, 6:18-21 (“It should be appreciated that the clocking architecture 112 is able to provided [sic] a <i>wide variety of clocking signal frequencies</i> to the different circuitry that it is couple to.”).
--	--

251. As discussed above in the evidence and disclosure table, Bertozzi discloses a singular clock signal that all the switches/routers and network interfaces (including the pipeline stages) (“the plurality of network stations”) and repeaters (M*N data storage elements) are responsive to a singular clock.

252. Also as discussed above in the evidence and disclosure table, Dielissen discloses that the routers/switch and network interfaces (“plurality of network station”) of Dielissen and the repeaters (“M*N data storage elements”) of Hassoun would be responsive to various clock signals.

253. A POSA would have been motivated, with a reasonable expectation of success, to include on-chip clocking circuitry (“a clock signal generator”), such as taught by Sullam, such that the plurality of network stations and the M*N data storage elements would be responsive to the clock signal generator. For Grounds 1 and 2, the on-chip clocking circuitry would have been a known and predictably

way to provide the single clock signal as taught by Bertozzi. For Ground 3, the on-chip clocking circuitry would have been a known way to provide “a wide variety of clocking signal frequencies to the different” to the different clock domains as required by Dielissen. Ex-1013, 6:18-21.

254. A POSA would have been motivated to use on-chip clocking circuitry (“clock signal generator”). **First**, as discussed in the table above, Dielissen and Bertozzi disclose the need for clocking on an integrated circuit. **Second**, including an on-chip clock signal generator allows for the entire SoC to operate without attaching an external clock signal generator. This simplifies the use of the chip as no additional hardware is required. **Third**, using an on-chip clock signal generator requires less total space on a board or in a multi-chip package. External clock signal generators require space outside of the chip (e.g. on the board or package) for both the clock signal generator itself and the connections to couple the external clock signal generator to the SoC.

255. A POSA would have had a reasonable expectation of success. **First**, it would have been routine and well within the knowledge of an ordinary artisan in the field. On-chip clocking circuitry was well known by the date of the purported invention disclosed by the '2893 patent. *See, e.g.*, Ex-1013. **Second**, the technology disclosed in Sullam is closely related to the technology disclosed by the references in Bertozzi and Dielissen. They all relate to on-chip technology,

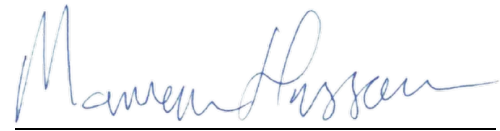
specifically on chip computers. A POSA would understand the Bertozzi teaches one or more computers on a chip. Bertozzi teaches an integrated circuit with several functional blocks integrated within including processors, co-processors, CPUs, DSPs, dedicated hardware accelerators, memory blocks, RAM, and I/O blocks. *E.g.*, Ex-1005, 2 (“SoCs represent high-complexity, high-value semiconductor products that incorporate building blocks from multiple sources (either in-house made or externally supplied): in particular, general-purpose fully programmable processors, co-processors, DSPs, dedicated hardware accelerators, memory blocks, I/O blocks, etc.”), Fig 1. Dielissen discloses multiple “IP modules” on a chip. Ex-1007, 1, 5, Fig. 11. A POSA would understand that IP modules include similar elements as discussed in Bertozzi, such as a CPU. Sullam discloses a computer on a chip. Ex-1013, 1:51-56 (“Specifically, the microcontroller is typically defined as a single chip (integrated circuit) that includes a processor, volatile memory such as Random Access Memory (RAM), non-volatile memory such as Read Only Memory (ROM), clocking source, and an input/output control unit. As such, the microcontroller is a computer on a chip.”).

Third, as discussed above, Bertozzi and Dielissen discloses the integration of various cores onto an integrated circuit. Ex-1005, 2; Ex-1007, 1. A POSA would have known, by early 2005, that understand that Sullam’s clocking circuitry 112 could be predictably integrated in Bertozzi or Dielissen as a core to generate the

needed clocks for Bertozzi and the combination of Dielissen and Hassoun in Ground 3. Given the state of knowledge in the field by 2005, doing this would have required only routine skill for a POSA.

I declare under penalty of perjury under the laws of the United States of America
that the foregoing is true and correct.

Dated: Dec 20, 2023



Dr. Marwan Hassoun

Appendix A

Reference	Claim Limitation, U.S. Patent No. 8,072,893
Independent Claim 10	
[10.pre]	A method of designing an integrated circuit comprising
[10.pre.a]	a plurality of functional blocks, and
[10.pre.b]	a data communication network comprising a plurality of network stations being interconnected via a plurality of communication channels for communicating data packages between the functional blocks,
[10.pre.c]	each data package comprising N data elements including a data element comprising routing information for the network stations, N being an integer of at least two,
[10.pre.d]	the plurality of network stations comprising a plurality of data routers and a plurality of network interfaces, each of the data routers being coupled to a functional block via a network interface;
[10.e]	the method comprising the acts of: identifying a first communication channel between a first network station and a second network station that has a data transfer delay exceeding a predefined delay threshold; and
[10.f]	in response to the identifying act, inserting M*N data storage elements into the data communication network, M being a positive integer, for introducing a delay of M*N cycles on the first communication channel.
Independent Claim 1	
[1.pre]	An integrated circuit comprising:
[1.a]	a plurality of functional blocks; and
[1.b]	a data communication network comprising a plurality of network stations being interconnected via a plurality of communication

	channels for communicating data packages between the functional blocks,
[1.c]	each data package comprising N data elements including a data element comprising routing information for the network stations, N being an integer of at least two,
[1.d]	the plurality of network stations comprising a plurality of data routers and a plurality of network interfaces, each of the data routers being coupled to a functional block via a network interface,
[1.e]	the data communication network comprising a first network station and a second network station interconnected through a first communication channel,
[1.f]	the data communication network further comprising M*N data storage elements, M being a positive integer, the data communication introducing a delay of M*N cycles on the first communication channel when the data communication network identifies the first communication channel as having a data transfer delay exceeding a predefined delay threshold.
Dependent Claim 2	
[2]	The integrated circuit as claimed in claim 1, wherein each communication channel of the plurality of communication channels comprises: a first subchannel between an input port of a first associated network station and an output port of a second associated network station; and a second subchannel between an input port of the second associated network station and an output port of the first associated network station for enabling duplex communication between the two associated network stations.
Dependent Claim 3	
[3]	The integrated circuit as claimed in claim 2, wherein the M*N data storage elements are distributed over the first subchannel and the second subchannel of the first communication channel.

Dependent Claim 6	
[6]	The integrated circuit as claimed in claim 1, wherein M=1.
Dependent Claim 8	
[8]	The integrated circuit as claimed in claim 1, further comprising a clock signal generator, the plurality of network stations and the M*N data storage elements being responsive to the clock signal generator.
Dependent Claim 9	
[9]	The integrated circuit as claimed in claim 1, wherein each network station comprises a first memory element and a second memory element associated with a communication port of said network station, the first memory element being arranged to store routing information from a data package requiring a first service level associated with said communication port and the second memory element being arranged to store routing information from a data package requiring a second service level associated with said communication port.
Independent Claim 7	
[7.pre]	An integrated circuit comprising:
[7.a]	a plurality of functional blocks; and
[7.b]	a data communication network comprising a plurality of network stations being interconnected via a plurality of communication channels for communicating data packages between the functional blocks,
[7.c]	each data package comprising N data elements including a data element comprising routing information for the network stations, N being an integer of at least two,
[7.d]	the plurality of network stations comprising a plurality of data routers and a plurality of network interfaces, each of the data routers being coupled to a functional block via a network interface,

[7.e]	the data communication network comprising a first network station and a second network station interconnected through a first communication channel,
[7.f]	the data communication network further comprising $M*N$ data storage elements, M being a positive integer, for introducing a delay of $M*N$ cycles on the first communication channel,
[7.g]	wherein each communication channel of the plurality of communication channels comprises: a first subchannel between an input port of a first associated network station and an output port of a second associated network station; and a second subchannel between an input port of the second associated network station and an output port of the first associated network station, and
[7.h]	wherein the first subchannel and the second subchannel of the first communication channel each comprise a delay introduced by $\frac{1}{2}*M*N$ data storage elements.