

Invalidity of U.S. Patent No. 8,086,800 in view of Radulescu III

A. Radulescu, et al., *An Efficient On-Chip Network Interface Offering Guaranteed Services, Shared-Memory Abstraction, and Flexible Network Configuration* (“Radulescu III”), which was published no later than March 2004 qualifies as prior art under at least 35 U.S.C. §§ 102 and 103 as to the asserted claim of U.S. Patent No. 8,086,800 (“’800 patent”), as discussed in greater detail in the chart that follows.¹

To the extent Plaintiff alleges that Radulescu III does not disclose any particular limitation of claim 12 (the “Asserted Claim”) of the ’800 patent, either expressly or inherently, it would have been obvious to a person of ordinary skill in the art as of the priority date of the ’800 patent to modify Radulescu III and/or to combine the teachings of Radulescu III with other prior art references, including but not limited to the prior art references presently found in Defendants’ Invalidity Contentions and any supplements thereto and the relevant section of charts for other prior art for the ’800 patent in a manner that would have rendered the Asserted Claim invalid as obvious. The inclusion of the particular limitation into the methods and systems described in Radulescu III would have been obvious for at least the following reasons: It would have been obvious to modify the methods and/or systems disclosed in Radulescu III, alone or in combination with the references identified in Defendants’ Invalidity Contentions, to meet this limitation based on: (a) the teachings in Radulescu III itself; (b) a person of ordinary skill in the art implementing a predictable variation of the methods and/or systems disclosed in Radulescu III or in the other identified references; (c) the logic, judgment, intelligence, creativity, innovation and common sense a person of ordinary skill in the art would employ; (d) the inferences and creative and routine steps that a person of ordinary skill in the art would employ; (e) market demand for methods and/or systems to include this limitation; and/or (f) design needs or market pressures to solve a problem with a finite number of solutions that could have produced the desired outcome.

Defendants reserve the right to amend or supplement this claim chart at a later date pursuant to the Docket Control Order issued by the Court on June 16, 2023 (Dkt. No. 59).

¹ Nothing in these claim charts should be construed as an admission regarding infringement, either literally or under the doctrine of equivalents, or as an admission regarding Defendants’ understanding of the proper scope of the Asserted Claim. Given Plaintiff’s vague and ambiguous infringement contentions, the exemplary citations herein necessarily account for a variety of possible infringement arguments and claim constructions. Defendants reserve the right to rely on additional citations or sources of evidence that also may be applicable, or that may become applicable in light of claim construction, changes in Plaintiff’s infringement contentions, and/or information obtained during discovery as the case progresses. Various positions put forth in this document are predicated on Plaintiff’s overly broad interpretation of its claims as evidenced by the infringement allegations received to date. Those positions are not intended to and do not necessarily reflect Defendants’ interpretation of the true and proper scope of Plaintiff’s claims, and Defendants reserve the right to adopt claim construction positions that differ from or even conflict with various positions put forth in this document.

A. Claim 12

'800 Patent Claim Limitation	Radulescu III
[12.pre] An integrated circuit comprising:	To the extent the preamble is limiting, Radulescu III discloses and/or renders obvious the preamble based on at least the following citations. See, e.g.: • Radulescu III at 1 In this paper we present a network interface for an on-chip network. Our network interface decouples computation from communication by offering a shared-memory abstraction, which is independent of the network implementation. We use a transaction-based protocol to achieve backward compatibility with existing bus protocols such as AXI, OCP and DTL. Our network interface has a modular architecture, which allows flexible instantiation. It provides both guaranteed and best-effort services via connections. These are configured via network interface ports using the network itself, instead of a separate control interconnect. An example instance of this network interface with 4 ports has an area of 0.143mm² in a 0:13µm technology, and runs at 500 MHz. • Radulescu III at 1 Networks on chip (NoC) have been proposed as a solution to the interconnect problem for highly complex chips [2, 3, 5, 9, 12, 14, 15, 17, 21, 27]. NoCs help designing chips in several ways: they (a) structure and manage wires in deep submicron technologies [2, 3, 9, 12, 21], (b) allow good wire utilization through sharing [5, 9, 12, 21], (c) scale better than buses [14, 21], (d) can be energy efficient and reliable [2, 5], and (e) decouple computation from communication through well-defined interfaces, enabling IP modules and interconnect to be designed in isolation, and to be integrated more easily [2, 13, 21, 24] • Radulescu III at 1 Our NI is intended for systems on chip (SoC), hence, it must have a low area. To enable the reuse of existing IP modules, we must provide a smooth transition from buses to NoCs. A shared-memory abstraction via transactions (e.g., read, write) ensures this. Further, we also have to provide a simple and flexible configuration, preferably using the NoC itself to avoid the need for a separate scalable interconnect.

'800 Patent Claim Limitation	Radulescu III
	To the extent Radulescu III does not expressly disclose the preamble, a person of ordinary skill in the art would have determined that the preamble is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Radulescu III. Further, one of ordinary skill in the art would have been motivated to modify Radulescu III or combine it with any of the present prior art references found in Defendant's Invalidity Contentions and any supplements thereto and the relevant section charts for other prior art for the '800 Patent in a manner that would result in the subject matter of the preamble given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Radulescu III.
[12.a] a plurality of processing modules including a first processing module having a first processing memory and a second processing module having a second processing memory;	Radulescu III discloses and/or renders obvious this limitation based on at least the following citations. See, e.g.: • Radulescu III at 1 Networks are composed of routers, which transport the data from one place to another, and network interfaces (NI), which implement the interface to the IP modules. In a previous article [21], we have shown the trade-offs in designing a cost-effective router combining guaranteed with best-effort traffic. In this paper, we focus on the other network component, the network interface. • Radulescu III at 1 We achieve a low-cost implementation of the NI by implementing the protocol stack in hardware, and by exploiting on-chip characteristics (such as the absence of transmission errors, relatively static configuration, tight synchronization) to implement only the relevant parts of a complete OSI stack. A hardware implementation of the protocol stack provides a much lower latency overhead compared to a software implementation. Further, a hardware implementation allows both hardware and software cores to be reused without change [4]. Our NI provides services at the transport layer in the ISO-OSI reference model [22], because this is the first layer where offered services are independent of the network implementation. This is a key ingredient in achieving the decoupling between computation and communication [16, 24], which allows IP modules and interconnect to be designed independently from each other. We provide transport-layer services by defining connections (e.g., point-to-point or multicast) configured for specific properties (e.g., throughput, ordering).

'800 Patent Claim Limitation	Radulescu III
	- Radulescu III at 1-2 We provide a modular NI, which can be configured at design time. This is, the number of ports and their type (i.e., configuration port, master port, or slave port), the number of connections at each port, memory allocated for the queues, the level of services per port, and the interface to the IP modules are all configurable at design (instantiation) time using an XML description [11]. The NI allows flexible NoC configuration at run time. Each connection can be configured individually, requiring configurable NoC components (i.e., router and NI). However, instead of using a separate control interconnect to program them, the NoC is used to program itself. This is performed through configuration ports using DTL-MMIO (memory-mapped IO) transactions [19]. The NoC can be configured in a distributed fashion (i.e., via multiple configuration ports), or centralized (i.e., via a single port). - Radulescu III at 2 As mentioned in the previous section, the communication services of the Æthereal NoC are defined to meet the following goals: (a) decouple computation (IP modules) from communication (NoC), (b) provide backward compatibility to existing bus protocols, (c) provide support for real-time communication, and (d) have a low-cost implementation. - Radulescu III at 2 The Æthereal NoC offers its services on connections, which can be point to point (one master, one slave), multicast (one master, multiple slaves, all slaves executing each transaction), and narrowcast (one master, multiple slaves, a transaction is executed by only one slave) [23]. Connections are composed of unidirectional point-to-point channels (between a single master and a single slave). To each channel, properties are attached, such as guaranteed message delivery or not, in order or unordered message delivery, and with or without timing guarantees. As a result, different properties can be attached to the request and response parts of a connection, or for different slaves within the same connection. Connections can be opened and closed at any time. Opening and closing of connections takes time, and is intended to be performed at a granularity larger than individual transactions. - Radulescu III at 2

'800 Patent Claim Limitation	Radulescu III
	Before the $\mathcal{A}$ethereal NoC can be used by an application, it must be configured. NoC (re)configuration means opening and closing connections in the system. Connections are set up depending on the application or the mode the system is running. Therefore, we must be able to open and close connections while the system is running. (Re)configuration can be partial or total (some or all connections are opened/closed, respectively). Opening a connection involves setting several registers, and allocating shared resources (for more details see Section 4). In the case of the current prototype of the $\mathcal{A}$ethereal NoC, for each pair of one master and one slave of a connection, there are 5 and 3 registers written at the master and slave network interfaces, respectively. The shared resources consist of the slots allocated to the connections. These slots can be configured using either a distributed or a centralized model. In the distributed case, a connection can be opened/closed from multiple network interface ports. Multiple configuration operations can be performed simultaneously, however, potential conflicts must also be solved (e.g., connection configurations initiated at two configuration ports may try to reserve the same slot in a router). Information about the slots is maintained in the routers, which also accept or reject a tentative slot allocation. • Radulescu III at 4 With the NI kernel described in the previous section, point-to-point connections (i.e., between on master and one slave) can be supported directly. These type of connections are useful in systems involving chains of modules communicating point to point with one another (e.g., video pixel processing [10]). For more complex types 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 1, we show a NI with two DTL and two AXI ports. All ports provide point-to-point 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. NoC instantiation is simple, as we use an XML description to automatically generate the VHDL code for the Nis as well as for the NoC topology. In Figure 3, we show an example of a narrowcast shell. Narrowcast connections are connections between one master and several slaves, where each transaction is executed by a single slave selected based on the address provided in the transaction [23]. Narrowcast connections provide a simple,

'800 Patent Claim Limitation	Radulescu III
	low-cost solution for a single shared address space mapped on multiple memories. It implements the splitting/merging of data going to/coming from these memories. • Radulescu III at 5 In Figure 9, 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 configuring the B to A connection at NI2, a connection from Cfg to NI2's CNIP must be set up. This connection is opened in two steps corresponding to the request and response channels. First, the request channel to the NI2's CNIP is set up by writing the necessary registers in NI1 (Step 1 in Figure 9). Second, we use this channel to set up (via the NoC) the response channel from NI2's CNIP to Cfg (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. See also, e.g., Section 4 (Network Interface Architecture); Figs. 1-9 To the extent Radulescu III does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Radulescu III. Further, one of ordinary skill in the art would have been motivated to modify Radulescu III or combine it with any of the present prior art references found in Defendant's Invalidity Contentions and any supplements thereto and the relevant section charts for other prior art for the '800 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Radulescu III.
[12.b] a plurality of interconnect modules including a first interconnect module and a second interconnect module, wherein the first processing module is connected to the	Radulescu III discloses and/or renders obvious this limitation based on at least the following citations. See, e.g.: • Radulescu III at 1 In this paper we present a network interface for an on-chip network. Our network interface decouples computation from communication by offering a shared-memory abstraction, which is independent of the network implementation. We use a transaction-based protocol to achieve

'800 Patent Claim Limitation	Radulescu III
first interconnect module and the second processing module is connected to the second interconnect module; and	backward compatibility with existing bus protocols such as AXI, OCP and DTL. Our network interface has a modular architecture, which allows flexible instantiation. It provides both guaranteed and best-effort services via connections. These are configured via network interface ports using the network itself, instead of a separate control interconnect. An example instance of this network interface with 4 ports has an area of 0.143mm^2 in a $0.13\mu\text{m}$ technology, and runs at 500 MHz. • Radulescu III at 1 Networks are composed of routers, which transport the data from one place to another, and network interfaces (NI), which implement the interface to the IP modules. In a previous article [21], we have shown the trade-offs in designing a cost-effective router combining guaranteed with best-effort traffic. In this paper, we focus on the other network component, the network interface. • Radulescu III at 1-2 We provide a modular NI, which can be configured at design time. This is, the number of ports and their type (i.e., configuration port, master port, or slave port), the number of connections at each port, memory allocated for the queues, the level of services per port, and the interface to the IP modules are all configurable at design (instantiation) time using an XML description [11]. The NI allows flexible NoC configuration at run time. Each connection can be configured individually, requiring configurable NoC components (i.e., router and NI). However, instead of using a separate control interconnect to program them, the NoC is used to program itself. This is performed through configuration ports using DTL-MMIO (memory-mapped IO) transactions [19]. The NoC can be configured in a distributed fashion (i.e., via multiple configuration ports), or centralized (i.e., via a single port). • Radulescu III at 2 As mentioned in the previous section, the communication services of the Æthereal NoC are defined to meet the following goals: (a) decouple computation (IP modules) from communication (NoC), (b) provide backward compatibility to existing bus protocols, (c) provide support for real-time communication, and (d) have a low-cost implementation. • Radulescu III at 2

'800 Patent Claim Limitation	Radulescu III
	The Æthereal NoC offers its services on connections, which can be point to point (one master, one slave), multicast (one master, multiple slaves, all slaves executing each transaction), and narrowcast (one master, multiple slaves, a transaction is executed by only one slave) [23]. Connections are composed of unidirectional point-to-point channels (between a single master and a single slave). To each channel, properties are attached, such as guaranteed message delivery or not, in order or unordered message delivery, and with or without timing guarantees. As a result, different properties can be attached to the request and response parts of a connection, or for different slaves within the same connection. Connections can be opened and closed at any time. Opening and closing of connections takes time, and is intended to be performed at a granularity larger than individual transactions. • Radulescu III at 2 Before the Æthereal NoC can be used by an application, it must be configured. NoC (re)configuration means opening and closing connections in the system. Connections are set up depending on the application or the mode the system is running. Therefore, we must be able to open and close connections while the system is running. (Re)configuration can be partial or total (some or all connections are opened/closed, respectively). Opening a connection involves setting several registers, and allocating shared resources (for more details see Section 4). In the case of the current prototype of the Æthereal NoC, for each pair of one master and one slave of a connection, there are 5 and 3 registers written at the master and slave network interfaces, respectively. The shared resources consist of the slots allocated to the connections. These slots can be configured using either a distributed or a centralized model. In the distributed case, a connection can be opened/closed from multiple network interface ports. Multiple configuration operations can be performed simultaneously, however, potential conflicts must also be solved (e.g., connection configurations initiated at two configuration ports may try to reserve the same slot in a router). Information about the slots is maintained in the routers, which also accept or reject a tentative slot allocation. • Radulescu III at 4 With the NI kernel described in the previous section, point-to-point connections (i.e., between on master and one slave) can be supported directly. These type of connections are useful in systems

'800 Patent Claim Limitation	Radulescu III
	involving chains of modules communicating point to point with one another (e.g., video pixel processing [10]). For more complex types 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 1, we show a NI with two DTL and two AXI ports. All ports provide point-to-point 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. NoC instantiation is simple, as we use an XML description to automatically generate the VHDL code for the Nis as well as for the NoC topology. In Figure 3, we show an example of a narrowcast shell. Narrowcast connections are connections between one master and several slaves, where each transaction is executed by a single slave selected based on the address provided in the transaction [23]. Narrowcast connections provide a simple, low-cost solution for a single shared address space mapped on multiple memories. It implements the splitting/merging of data going to/coming from these memories. • Radulescu III at 5 In Figure 9, 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 configuring the B to A connection at NI2, a connection from Cfg to NI2's CNIP must be set up. This connection is opened in two steps corresponding to the request and response channels. First, the request channel to the NI2's CNIP is set up by writing the necessary registers in NI1 (Step 1 in Figure 9). Second, we use this channel to set up (via the NoC) the response channel from NI2's CNIP to Cfg (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. See also, e.g., Section 4 (Network Interface Architecture); Figs. 1-9 To the extent Radulescu III does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Radulescu III. Further, one of ordinary skill in the

'800 Patent Claim Limitation	Radulescu III
	art would have been motivated to modify Radulescu III or combine it with any of the present prior art references found in Defendant's Invalidity Contentions and any supplements thereto and the relevant section charts for other prior art for the '800 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Radulescu III.
[12.c] a network for connecting the first interconnect module to the second interconnect module in order to provide a communication connection between the first processing module and the second processing module, wherein the first interconnect module is connected between the first processing module and the network, and wherein the second interconnect module is connected between the second processing module and the network;	Radulescu III discloses and/or renders obvious this limitation based on at least the following citations. See, e.g.: • Radulescu III at 1 In this paper we present a network interface for an on-chip network. Our network interface decouples computation from communication by offering a shared-memory abstraction, which is independent of the network implementation. We use a transaction-based protocol to achieve backward compatibility with existing bus protocols such as AXI, OCP and DTL. Our network interface has a modular architecture, which allows flexible instantiation. It provides both guaranteed and best-effort services via connections. These are configured via network interface ports using the network itself, instead of a separate control interconnect. An example instance of this network interface with 4 ports has an area of 0.143mm² in a 0:13μm technology, and runs at 500 MHz. • Radulescu III at 1 Networks are composed of routers, which transport the data from one place to another, and network interfaces (NI), which implement the interface to the IP modules. In a previous article [21], we have shown the trade-offs in designing a cost-effective router combining guaranteed with best-effort traffic. In this paper, we focus on the other network component, the network interface. • Radulescu III at 1-2 We provide a modular NI, which can be configured at design time. This is, the number of ports and their type (i.e., configuration port, master port, or slave port), the number of connections at each port, memory allocated for the queues, the level of services per port, and the interface to the IP modules are all configurable at design (instantiation) time using an XML description [11]. The NI allows flexible NoC configuration at run time. Each connection can be configured individually, requiring configurable NoC components (i.e., router and NI). However, instead of

'800 Patent Claim Limitation	Radulescu III
	using a separate control interconnect to program them, the NoC is used to program itself. This is performed through configuration ports using DTL-MMIO (memory-mapped IO) transactions [19]. The NoC can be configured in a distributed fashion (i.e., via multiple configuration ports), or centralized (i.e., via a single port). • Radulescu III at 2 As mentioned in the previous section, the communication services of the Æthereal NoC are defined to meet the following goals: (a) decouple computation (IP modules) from communication (NoC), (b) provide backward compatibility to existing bus protocols, (c) provide support for real-time communication, and (d) have a low-cost implementation. • Radulescu III at 2 The Æthereal NoC offers its services on connections, which can be point to point (one master, one slave), multicast (one master, multiple slaves, all slaves executing each transaction), and narrowcast (one master, multiple slaves, a transaction is executed by only one slave) [23]. Connections are composed of unidirectional point-to-point channels (between a single master and a single slave). To each channel, properties are attached, such as guaranteed message delivery or not, in order or unordered message delivery, and with or without timing guarantees. As a result, different properties can be attached to the request and response parts of a connection, or for different slaves within the same connection. Connections can be opened and closed at any time. Opening and closing of connections takes time, and is intended to be performed at a granularity larger than individual transactions. • Radulescu III at 2 Before the Æthereal NoC can be used by an application, it must be configured. NoC (re)configuration means opening and closing connections in the system. Connections are set up depending on the application or the mode the system is running. Therefore, we must be able to open and close connections while the system is running. (Re)configuration can be partial or total (some or all connections are opened/closed, respectively). Opening a connection involves setting several registers, and allocating shared resources (for more details see Section 4). In the case of the current prototype of the Æthereal NoC, for each pair of one

'800 Patent Claim Limitation	Radulescu III
	master and one slave of a connection, there are 5 and 3 registers written at the master and slave network interfaces, respectively. The shared resources consist of the slots allocated to the connections. These slots can be configured using either a distributed or a centralized model. In the distributed case, a connection can be opened/closed from multiple network interface ports. Multiple configuration operations can be performed simultaneously, however, potential conflicts must also be solved (e.g., connection configurations initiated at two configuration ports may try to reserve the same slot in a router). Information about the slots is maintained in the routers, which also accept or reject a tentative slot allocation. • Radulescu III at 4 With the NI kernel described in the previous section, point-to-point connections (i.e., between on master and one slave) can be supported directly. These type of connections are useful in systems involving chains of modules communicating point to point with one another (e.g., video pixel processing [10]). For more complex types 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 1, we show a NI with two DTL and two AXI ports. All ports provide point-to-point 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. NoC instantiation is simple, as we use an XML description to automatically generate the VHDL code for the Nis as well as for the NoC topology. In Figure 3, we show an example of a narrowcast shell. Narrowcast connections are connections between one master and several slaves, where each transaction is executed by a single slave selected based on the address provided in the transaction [23]. Narrowcast connections provide a simple, low-cost solution for a single shared address space mapped on multiple memories. It implements the splitting/merging of data going to/coming from these memories. • Radulescu III at 5 In Figure 9, 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 configuring the B to A connection at NI2, a connection

'800 Patent Claim Limitation	Radulescu III
	from Cfg to NI2's CNIP must be set up. This connection is opened in two steps corresponding to the request and response channels. First, the request channel to the NI2's CNIP is set up by writing the necessary registers in NI1 (Step 1 in Figure 9). Second, we use this channel to set up (via the NoC) the response channel from NI2's CNIP to Cfg (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. See also, e.g., Section 4 (Network Interface Architecture); Figs. 1-9 To the extent Radulescu III does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Radulescu III. Further, one of ordinary skill in the art would have been motivated to modify Radulescu III or combine it with any of the present prior art references found in Defendant's Invalidity Contentions and any supplements thereto and the relevant section charts for other prior art for the '800 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Radulescu III.
[12.d] wherein the first interconnect module includes a first memory for buffering first data from the first processing module, the first interconnect module further including a first determination unit, wherein the first determination unit is configured to determine a first optimal amount of data to be buffered by the first memory, and the second interconnect module includes	Radulescu III discloses and/or renders obvious this limitation based on at least the following citations. See, e.g.: • Radulescu III at 2 The Æthereal NoC offers its services on connections, which can be point to point (one master, one slave), multicast (one master, multiple slaves, all slaves executing each transaction), and narrowcast (one master, multiple slaves, a transaction is executed by only one slave) [23]. Connections are composed of unidirectional point-to-point channels (between a single master and a single slave). To each channel, properties are attached, such as guaranteed message delivery or not, in order or unordered message delivery, and with or without timing guarantees. As a result, different properties can be attached to the request and response parts of a connection, or for different slaves within the same connection. Connections can be opened and closed at any time. Opening and closing of connections takes time, and is intended to be performed at a granularity larger than individual transactions.

'800 Patent Claim Limitation	Radulescu III
a second memory for buffering second data from the second processing module, the second interconnect module further including a second determination unit, wherein the second determination unit is configured to determine a second optimal amount of data to be buffered by the second memory,	Support for real-time communication is achieved by providing throughput, latency and jitter guarantees. In <i>Æthereal</i>, this is implemented by configuring connections as pipelined time-division-multiplexed circuits over the network. Time multiplexing is only possible when the network routers have a notion of synchronicity which allows slots to be reserved consecutively in a sequence of routers [13, 21]. This scheme [21] has smaller packet buffers, and, hence, has lower implementation cost compared to alternatives, such as rate-based packet switching [29], or deadline-based packet switching [20]. • Radulescu III at 3 In the NI kernel, there are two message queues for each point-to-point connection (one source queue, for messages going to the NoC, and one destination queue, for messages coming from the NoC). Their size is also selected at the NI instantiation time. In our NI, queues are implemented using custom-made hardware fifos, and are also used to provide the clock domain crossing between the network and the IP modules. Each port can, therefore, have a different clock frequency. Each channel is configured individually. In a first prototype of the <i>Æthereal</i> NI, we can configure if a channel provides time guarantees (GT) or not (we call this best effort, BE), reserve slots for GT connections, configure the end-to-end flow control, and the routing information. 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 [26]. For each channel, there is a counter (Space) tracking the empty buffer space of the remote destination queue. This counter is initialized 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. In the <i>Æthereal</i> prototype, we piggyback credits in the header of the packets for the data in the other direction to improve NoC efficiency. Note that at most Space data items can be transmitted before credits are received. We call the minimum between the data items in the queue and the value in the counter Space, the sendable data. • Radulescu III at 3-4 There are multiple channels which may require data transmission, we implement a scheduler to arbitrate between them. The scheduler checks if the current slot is reserved for a GT channel. If the

'800 Patent Claim Limitation	Radulescu III
	slot is reserved, is the GT channel has data which can be transmitted, and if there is space in the channel's destination buffer, then the channel is granted data transmission. Otherwise, the scheduler selects a BE channel with data and remote space using some arbitration scheme: e.g. round-robin, weighted roundrobin, or based on the queue filling. To optimize the NoC utilization, it is preferable to send longer packets. To achieve this, we implemented a configurable threshold mechanism, which skips a channel as long as the sendable data is below the threshold. This is applicable for both BE and GT channels. To prevent starvation at user/application level (e.g., due to write data being buffered indefinitely on which the IP module waits for an acknowledge), we also provide a flush signal for each channel (and a bit in the message header) to temporarily override the threshold. When the flush signal is high for a cycle, a snapshot of its source queue filling is taken, and as long as all the words in the queue at the time of flushing have not been sent, the threshold for that queue is bypassed. • Radulescu III at 4 A similar threshold is set for credit transmission. The reason is that, when there is no data on which the credits can be piggybacked, the credits are sent as empty packets, thus, consuming extra bandwidth. To minimize the bandwidth consumed by credits, a credit threshold is set, which allows credits to be transmitted only when their sum is above the threshold. Similarly to the data case, to prevent possible starvation, we provide a flush signal to force credits to be sent even when they are below their threshold. As credits are piggybacked on packets, a queue becomes eligible for scheduling when either the amount of sendable data are above a first threshold, or when the amount of credits is above a second threshold. However, once a queue is selected, a packet containing the largest possible amount of credits and data will be produced. Note the amount of credits is bound by implementation to the given number of bits in the packet header, and packet have a maximum length to avoid links being used exclusively by a packet/channel, which would cause congestion. • Radulescu III at 4 When a slave using a connectionless protocol (e.g., DTL) is connected to a NI port supporting multiple connections, a multiconnection shell must be included to arbitrate between the connections. A multi-connection shell (see Figure 4) includes a scheduler to select connections from which

'800 Patent Claim Limitation	Radulescu III
	messages are consumed, based e.g., on their filling. As for the narrowcast, the multi-connection shell has a connection id history for scheduling the responses. See also, e.g., Section 4 (Network Interface Architecture); Figs. 1-9 To the extent Radulescu III does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Radulescu III. Further, one of ordinary skill in the art would have been motivated to modify Radulescu III or combine it with any of the present prior art references found in Defendant's Invalidity Contentions and any supplements thereto and the relevant section charts for other prior art for the '800 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Radulescu III.
[12.e] the first interconnect module transferring the first data to the second processing module when the first data buffered in the first memory reaches the first optimal amount, and the second interconnect module transferring the second data to the first processing module when the second data buffered in the second memory reaches the second optimal amount	Radulescu III discloses and/or renders obvious this limitation based on at least the following citations. See, e.g.: • Radulescu III at 2 The Æthereal NoC offers its services on connections, which can be point to point (one master, one slave), multicast (one master, multiple slaves, all slaves executing each transaction), and narrowcast (one master, multiple slaves, a transaction is executed by only one slave) [23]. Connections are composed of unidirectional point-to-point channels (between a single master and a single slave). To each channel, properties are attached, such as guaranteed message delivery or not, in order or unordered message delivery, and with or without timing guarantees. As a result, different properties can be attached to the request and response parts of a connection, or for different slaves within the same connection. Connections can be opened and closed at any time. Opening and closing of connections takes time, and is intended to be performed at a granularity larger than individual transactions. Support for real-time communication is achieved by providing throughput, latency and jitter guarantees. In Æthereal, this is implemented by configuring connections as pipelined time-division-multiplexed circuits over the network. Time multiplexing is only possible when the network routers have a notion of synchronicity which allows slots to be reserved consecutively in a sequence of

'800 Patent Claim Limitation	Radulescu III
	routers [13, 21]. This scheme [21] has smaller packet buffers, and, hence, has lower implementation cost compared to alternatives, such as rate-based packet switching [29], or deadline-based packet switching [20]. - Radulescu III at 3 In the NI kernel, there are two message queues for each point-to-point connection (one source queue, for messages going to the NoC, and one destination queue, for messages coming from the NoC). Their size is also selected at the NI instantiation time. In our NI, queues are implemented using custom-made hardware fifos, and are also used to provide the clock domain crossing between the network and the IP modules. Each port can, therefore, have a different clock frequency. Each channel is configured individually. In a first prototype of the Æthereal NI, we can configure if a channel provides time guarantees (GT) or not (we call this best effort, BE), reserve slots for GT connections, configure the end-to-end flow control, and the routing information. 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 [26]. For each channel, there is a counter (Space) tracking the empty buffer space of the remote destination queue. This counter is initialized 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. In the Æthereal prototype, we piggyback credits in the header of the packets for the data in the other direction to improve NoC efficiency. Note that at most Space data items can be transmitted before credits are received. We call the minimum between the data items in the queue and the value in the counter Space, the sendable data. - Radulescu III at 3-4 There are multiple channels which may require data transmission, we implement a scheduler to arbitrate between them. The scheduler checks if the current slot is reserved for a GT channel. If the slot is reserved, is the GT channel has data which can be transmitted, and if there is space in the channel's destination buffer, then the channel is granted data transmission. Otherwise, the scheduler selects a BE channel with data and remote space using some arbitration scheme: e.g. round-robin, weighted roundrobin, or based on the queue filling.

'800 Patent Claim Limitation	Radulescu III
	To optimize the NoC utilization, it is preferable to send longer packets. To achieve this, we implemented a configurable threshold mechanism, which skips a channel as long as the sendable data is below the threshold. This is applicable for both BE and GT channels. To prevent starvation at user/application level (e.g., due to write data being buffered indefinitely on which the IP module waits for an acknowledge), we also provide a flush signal for each channel (and a bit in the message header) to temporarily override the threshold. When the flush signal is high for a cycle, a snapshot of its source queue filling is taken, and as long as all the words in the queue at the time of flushing have not been sent, the threshold for that queue is bypassed. • Radulescu III at 4 A similar threshold is set for credit transmission. The reason is that, when there is no data on which the credits can be piggybacked, the credits are sent as empty packets, thus, consuming extra bandwidth. To minimize the bandwidth consumed by credits, a credit threshold is set, which allows credits to be transmitted only when their sum is above the threshold. Similarly to the data case, to prevent possible starvation, we provide a flush signal to force credits to be sent even when they are below their threshold. As credits are piggybacked on packets, a queue becomes eligible for scheduling when either the amount of sendable data are above a first threshold, or when the amount of credits is above a second threshold. However, once a queue is selected, a packet containing the largest possible amount of credits and data will be produced. Note the amount of credits is bound by implementation to the given number of bits in the packet header, and packet have a maximum length to avoid links being used exclusively by a packet/channel, which would cause congestion. • Radulescu III at 4 When a slave using a connectionless protocol (e.g., DTL) is connected to a NI port supporting multiple connections, a multiconnection shell must be included to arbitrate between the connections. A multi-connection shell (see Figure 4) includes a scheduler to select connections from which messages are consumed, based e.g., on their filling. As for the narrowcast, the multi-connection shell has a connection id history for scheduling the responses. See also, e.g., Section 4 (Network Interface Architecture); Figs. 1-9

'800 Patent Claim Limitation	Radulescu III
	To the extent Radulescu III does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Radulescu III. Further, one of ordinary skill in the art would have been motivated to modify Radulescu III or combine it with any of the present prior art references found in Defendant's Invalidity Contentions and any supplements thereto and the relevant section charts for other prior art for the '800 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Radulescu III.
[12.f] wherein at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave,	Radulescu III discloses and/or renders obvious this limitation based on at least the following citations. See, e.g.: • Radulescu III at 1 We offer guaranteed services as they are essential for a compositional construction (design and programming) of SoC. The reasons are that they limit the possible interactions of IPs with the communication environment [12, 13], separate the IP requirements and their implementation, and make application quality of service independent of the IP and NoC implementations. Examples of such guarantees are lower bounds on throughput, and upper bounds on latency. • Radulescu III at 2 The Æthereal NoC offers its services on connections, which can be point to point (one master, one slave), multicast (one master, multiple slaves, all slaves executing each transaction), and narrowcast (one master, multiple slaves, a transaction is executed by only one slave) [23]. Connections are composed of unidirectional point-to-point channels (between a single master and a single slave). To each channel, properties are attached, such as guaranteed message delivery or not, in order or unordered message delivery, and with or without timing guarantees. As a result, different properties can be attached to the request and response parts of a connection, or for different slaves within the same connection. Connections can be opened and closed at any time. Opening and closing of connections takes time, and is intended to be performed at a granularity larger than individual transactions.

'800 Patent Claim Limitation	Radulescu III
	Support for real-time communication is achieved by providing throughput, latency and jitter guarantees. In <i>Æthereal</i>, this is implemented by configuring connections as pipelined time-division-multiplexed circuits over the network. Time multiplexing is only possible when the network routers have a notion of synchronicity which allows slots to be reserved consecutively in a sequence of routers [13, 21]. This scheme [21] has smaller packet buffers, and, hence, has lower implementation cost compared to alternatives, such as rate-based packet switching [29], or deadline-based packet switching [20]. Throughput guarantees are given by the number of slots reserved for a connection. Slots correspond to a given bandwidth: B_i, and, therefore, reserving N slots for a connection results in a total bandwidth of $N \times B_i$. The latency bound is given by the waiting time until the reserved slot arrives and the number of routers data passes to reach its destination. Jitter is given by the maximum distance between two slot reservations. • Radulescu III at 2-3 In the distributed case, a connection can be opened/closed from multiple network interface ports. Multiple configuration operations can be performed simultaneously, however, potential conflicts must also be solved (e.g., connection configurations initiated at two configuration ports may try to reserve the same slot in a router). Information about the slots is maintained in the routers, which also accept or reject a tentative slot allocation. In a centralized system, there is only one place that performs NoC configuration. In such a case, the slot information can be stored in the configuration module instead of the routers, which simplifies the design, and, in the case of small NoCs, may even speed up configuration. For large NoCs, however, centralized configuration can introduce a bottleneck. • Radulescu III at 3-4 There are multiple channels which may require data transmission, we implement a scheduler to arbitrate between them. The scheduler checks if the current slot is reserved for a GT channel. If the slot is reserved, is the GT channel has data which can be transmitted, and if there is space in the channel's destination buffer, then the channel is granted data transmission. Otherwise, the scheduler selects a BE channel with data and remote space using some arbitration scheme: e.g. round-robin, weighted roundrobin, or based on the queue filling.

'800 Patent Claim Limitation	Radulescu III
	To optimize the NoC utilization, it is preferable to send longer packets. To achieve this, we implemented a configurable threshold mechanism, which skips a channel as long as the sendable data is below the threshold. This is applicable for both BE and GT channels. To prevent starvation at user/application level (e.g., due to write data being buffered indefinitely on which the IP module waits for an acknowledge), we also provide a flush signal for each channel (and a bit in the message header) to temporarily override the threshold. When the flush signal is high for a cycle, a snapshot of its source queue filling is taken, and as long as all the words in the queue at the time of flushing have not been sent, the threshold for that queue is bypassed. • Radulescu III at 4 As credits are piggybacked on packets, a queue becomes eligible for scheduling when either the amount of sendable data are above a first threshold, or when the amount of credits is above a second threshold. However, once a queue is selected, a packet containing the largest possible amount of credits and data will be produced. Note the amount of credits is bound by implementation to the given number of bits in the packet header, and packet have a maximum length to avoid links being used exclusively by a packet/channel, which would cause congestion. • Radulescu III at 4 When a slave using a connectionless protocol (e.g., DTL) is connected to a NI port supporting multiple connections, a multiconnection shell must be included to arbitrate between the connections. A multi-connection shell (see Figure 4) includes a scheduler to select connections from which messages are consumed, based e.g., on their filling. As for the narrowcast, the multi-connection shell has a connection id history for scheduling the responses. See also, e.g., Section 2 (NoC Services); Section 4 (Network Interface Architecture); Figs. 1-9 To the extent Radulescu III does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Radulescu III. Further, one of ordinary skill in the art would have been motivated to modify Radulescu III or combine it with any of the present prior art references found in Defendant's Invalidity Contentions and any supplements thereto and the

'800 Patent Claim Limitation	Radulescu III
	relevant section charts for other prior art for the '800 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Radulescu III.
[12.g] wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.	Radulescu III discloses and/or renders obvious this limitation based on at least the following citations. See, e.g.: • Radulescu III at 1 We offer guaranteed services as they are essential for a compositional construction (design and programming) of SoC. The reasons are that they limit the possible interactions of IPs with the communication environment [12, 13], separate the IP requirements and their implementation, and make application quality of service independent of the IP and NoC implementations. Examples of such guarantees are lower bounds on throughput, and upper bounds on latency. • Radulescu III at 2 The <i>Æthereal</i> NoC offers its services on connections, which can be point to point (one master, one slave), multicast (one master, multiple slaves, all slaves executing each transaction), and narrowcast (one master, multiple slaves, a transaction is executed by only one slave) [23]. Connections are composed of unidirectional point-to-point channels (between a single master and a single slave). To each channel, properties are attached, such as guaranteed message delivery or not, in order or unordered message delivery, and with or without timing guarantees. As a result, different properties can be attached to the request and response parts of a connection, or for different slaves within the same connection. Connections can be opened and closed at any time. Opening and closing of connections takes time, and is intended to be performed at a granularity larger than individual transactions. Support for real-time communication is achieved by providing throughput, latency and jitter guarantees. In <i>Æthereal</i>, this is implemented by configuring connections as pipelined time-division-multiplexed circuits over the network. Time multiplexing is only possible when the network routers have a notion of synchronicity which allows slots to be reserved consecutively in a sequence of routers [13, 21]. This scheme [21] has smaller packet buffers, and, hence, has lower implementation cost compared to alternatives, such as rate-based packet switching [29], or deadline-based packet switching [20].

'800 Patent Claim Limitation	Radulescu III
	Throughput guarantees are given by the number of slots reserved for a connection. Slots correspond to a given bandwidth: B_i, and, therefore, reserving N slots for a connection results in a total bandwidth of $N \times B_i$. The latency bound is given by the waiting time until the reserved slot arrives and the number of routers data passes to reach its destination. Jitter is given by the maximum distance between two slot reservations. • Radulescu III at 2-3 In the distributed case, a connection can be opened/closed from multiple network interface ports. Multiple configuration operations can be performed simultaneously, however, potential conflicts must also be solved (e.g., connection configurations initiated at two configuration ports may try to reserve the same slot in a router). Information about the slots is maintained in the routers, which also accept or reject a tentative slot allocation. In a centralized system, there is only one place that performs NoC configuration. In such a case, the slot information can be stored in the configuration module instead of the routers, which simplifies the design, and, in the case of small NoCs, may even speed up configuration. For large NoCs, however, centralized configuration can introduce a bottleneck. • Radulescu III at 3-4 There are multiple channels which may require data transmission, we implement a scheduler to arbitrate between them. The scheduler checks if the current slot is reserved for a GT channel. If the slot is reserved, is the GT channel has data which can be transmitted, and if there is space in the channel's destination buffer, then the channel is granted data transmission. Otherwise, the scheduler selects a BE channel with data and remote space using some arbitration scheme: e.g. round-robin, weighted roundrobin, or based on the queue filling. To optimize the NoC utilization, it is preferable to send longer packets. To achieve this, we implemented a configurable threshold mechanism, which skips a channel as long as the sendable data is below the threshold. This is applicable for both BE and GT channels. To prevent starvation at user/application level (e.g., due to write data being buffered indefinitely on which the IP module waits for an acknowledge), we also provide a flush signal for each channel (and a bit in the message header) to temporarily override the threshold. When the flush signal is high for a cycle, a snapshot

'800 Patent Claim Limitation	Radulescu III
	of its source queue filling is taken, and as long as all the words in the queue at the time of flushing have not been sent, the threshold for that queue is bypassed. • Radulescu III at 4 As credits are piggybacked on packets, a queue becomes eligible for scheduling when either the amount of sendable data are above a first threshold, or when the amount of credits is above a second threshold. However, once a queue is selected, a packet containing the largest possible amount of credits and data will be produced. Note the amount of credits is bound by implementation to the given number of bits in the packet header, and packet have a maximum length to avoid links being used exclusively by a packet/channel, which would cause congestion. • Radulescu III at 4 When a slave using a connectionless protocol (e.g., DTL) is connected to a NI port supporting multiple connections, a multiconnection shell must be included to arbitrate between the connections. A multi-connection shell (see Figure 4) includes a scheduler to select connections from which messages are consumed, based e.g., on their filling. As for the narrowcast, the multi-connection shell has a connection id history for scheduling the responses. See also, e.g., Section 2 (NoC Services); Section 4 (Network Interface Architecture); Figs. 1-9 To the extent Radulescu III does not expressly disclose this limitation, a person of ordinary skill in the art would have determined that this limitation is either inherent and/or obvious to one of ordinary skill in the art in view of the teachings of Radulescu III. Further, one of ordinary skill in the art would have been motivated to modify Radulescu III or combine it with any of the present prior art references found in Defendant's Invalidity Contentions and any supplements thereto and the relevant section charts for other prior art for the '800 Patent in a manner that would result in the subject matter of this limitation given, at the very least, the person's understanding of the state of the art, the problems addressed and solved in the prior art, and the teachings of Radulescu III.