Exhibit 596-10

Invalidity of the ’596 Patent by Khawand ’528 and Other Prior Art

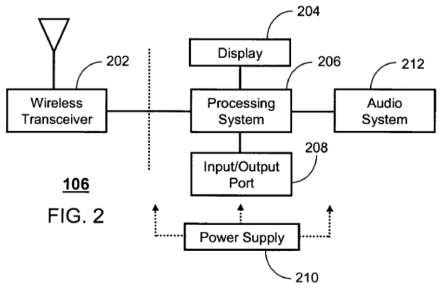
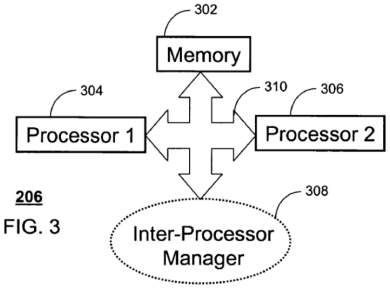
The asserted claims of U.S. Patent No. 8,924,596 are anticipated and/or rendered obvious in view of US7398528B2 (“Khawand ’528”) alone or in combination with knowledge of one of skill in the art or other prior art.

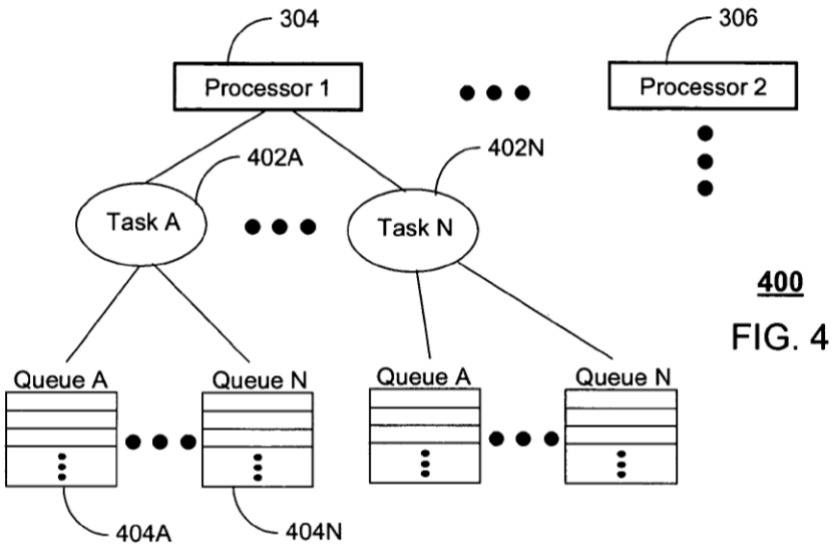
This claim chart is based on Defendants’ present understanding of the asserted claims, Plaintiffs’ vague and overbroad infringement contentions (and corresponding, implicit claim constructions), and Defendants’ investigation to date. Defendants are not adopting Plaintiffs’ apparent constructions, nor are Defendants admitting to the correctness of any particular construction. Defendants reserve all rights to amend this invalidity claim chart in light of any claim construction developments or any amendments to Plaintiffs’ infringement contentions. Further, as discovery is ongoing and Defendants continue to seek discovery from third parties regarding the references identified in Defendants’ invalidity contentions as well as other potential prior art, Defendants reserve the right to amend or supplement this claim chart in connection with Defendants’ continuing investigation. At a minimum, Defendants intend to revise these invalidity contentions as appropriate based on discovery from third parties as discovery progresses.

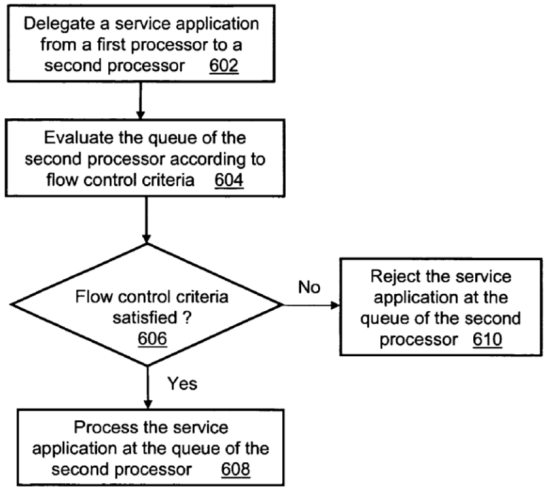
The claim chart below identifies where each limitation of each asserted claim of the ’596 patent can be found in Khawand ’528. This analysis is based on Plaintiff’s interpretation of the claims as indicated in their Infringement Contentions. The citations provided below are exemplary, rather than exhaustive, and Defendants reserve the right to rely upon any other portion of the cited references.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
[1pre] A system for dividing and synchronizing a processing task across a plurality of processing elements comprising:	To the extent the preamble is considered a claim limitation, Khawand ’528 teaches “[a] system for dividing and synchronizing a processing task across a plurality of processing elements,” including at least the following exemplary citations. For example, Khawand ’528 teaches a system for dividing and synchronizing processing tasks by disclosing, as non–limiting examples, a multiprocessor system including an inter–processor manager that divides and synchronizes processing tasks. *See, e.g.:* “A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at Abstract.

IPR2025-00486
Concurrent Ventures LLC, XtreamEdge Inc. EX2029

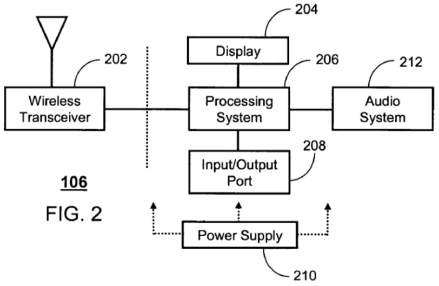
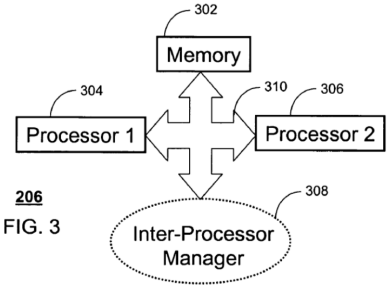
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	  Khawand ’528 at Figures 2 and 3.

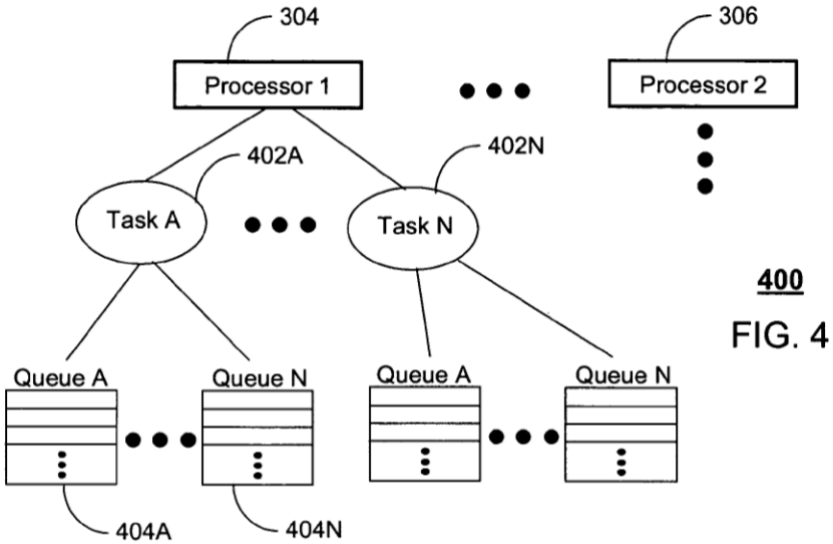
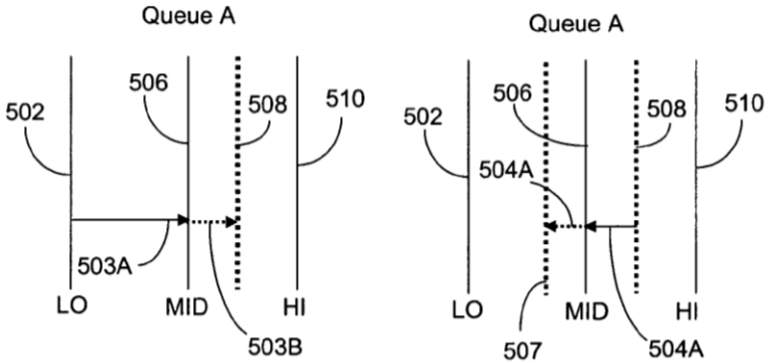
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figure 4.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 FIG. 6 600 Khawand ’528 at Figure 6. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 35–48. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "The processing system 206 is the processing center for the mobile device 106, which controls operations of the foregoing components 202-212. The processing system 206 controls singly or concurrently voice processing (e.g., voice calls), multi-media processing (video MPEG 4 player), data message processing (e.g., application downloads, short message system exchanges, etc.), and other house-keeping functions of the mobile device 106. The processing system 206 uses an operating system for distributing service applications in a manner consistent with method 600 of FIG. 6. Service applications can represent all inclusive software algorithms (e.g., an MPEG 4 video decoder), or portions thereof subdivided into tasks (e.g., input/output drivers, display drivers, etc.) and/or primitive instructions such as microcode." Khawand '528 at col. 3, ll. 4–17. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49.
[1a] an input queue implemented in hardware;	Khawand '528 teaches "an input queue implemented in hardware," including at least the following exemplary citations. Plaintiffs have cited the use of an input queue implemented in memory through software as allegedly satisfying limitation [1a]. *See* Am. Compl. at ¶59; '596 Infringement Contentions at 43–58. To the extent that

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	Plaintiffs argue that Khawand ’528’s input queue is in memory Khawand ’528 teaches or renders obvious claim limitation [1a] under Plaintiffs’ apparent interpretation of the claim. For example, Khawand ’528 teaches an input queue implemented in hardware, by disclosing, as non–limiting examples, task queues. *See, e.g.:* Defendants have proposed a construction that “queue implemented in hardware” means “queue implemented in physical circuits without software intervention.” The input queue in Khawand ’528 meets Defendants’ construction because, for example, Khawand ’528 teaches a hardware implementation of its teachings, which includes its teachings of queues. *See, e.g.*, Khawand ’528 at 6:14–17 (“In light of the foregoing description, it should be recognized that embodiments in the present invention could be realized in hardware, software, or a combination of hardware and software.”). “A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at Abstract.

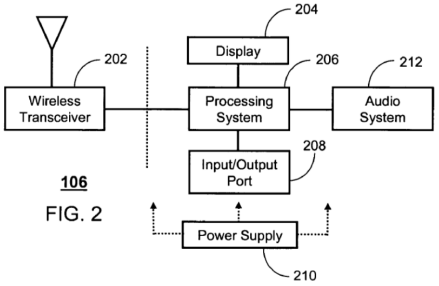
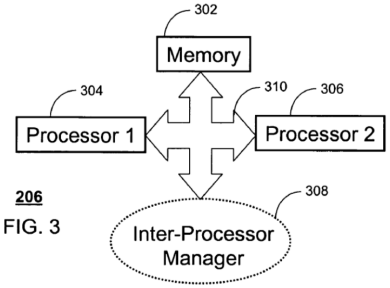
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	  Khawand ’528 at Figures 2 and 3.

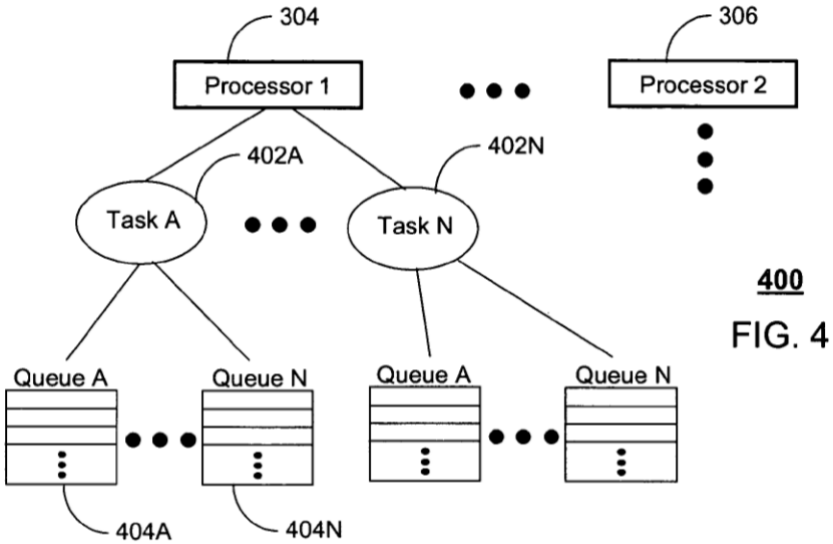
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figure 4.  FIG. 5 **500** Khawand ’528 at Figure 5. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 35–48. "In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "FIG. 3 shows a detailed view of the processing system 206, comprising a conventional memory 302, a plurality of processors 304-306 (shown by way of example as first and second processors) and an inter-processor manager 308 according to an embodiment of the present invention. The memory 302 can be a single or a combination of conventional memory devices (e.g., dynamic random access memory, flash memory, etc.) operated on individually or jointly by the elements 304-308 of the processing system 206. The plurality of processors 304-306 employ conventional processing elements such as, for example, an ARM processor. The inter-processor manager 308 can comprise a conventional application specific integrated circuit (ASIC), a microprocessor, a state machine, or other processing means that is capable of operating according to the method 600 of FIG. 6. Alternatively, the processing function of the inter-processor manager

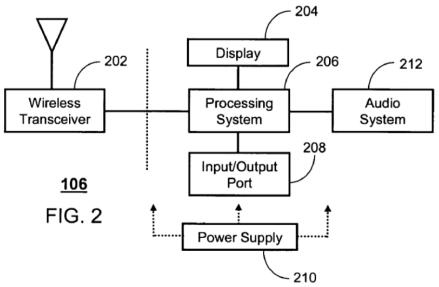
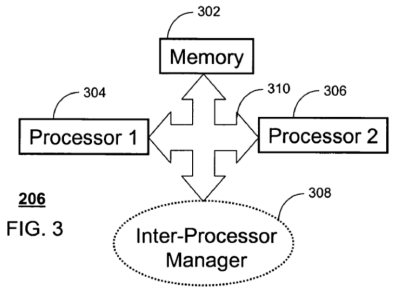
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	308 can take the form of hardware, software or combinations thereof operating in whole or in part from the first or second processors 304, or independent hardware. The processing elements 302-308 of the processing system 206 are interconnected by a conventional communication bus 310. The construction of the bus 310 can be physical or logical. In the former, the physical connection can be a conventional serial or parallel bus for transmitting signals between devices sourced by a conventional transceiver (e.g., universal asynchronous receiver transmitter—UART, or universal serial bus—USB driver). Alternatively, a portion or whole of the communication bus 310 can represent a logical connection such as software components sharing data structures. The processing elements 302-308 coupled to the communication bus 310 can be represented in a single integrated circuit (IC) device with sub-elements operating therein, or a combination of ICs, each operating as a corresponding one of the elements 302-308 of the processing system 206." Khawand '528 at col. 3, ll. 18–51. "FIG. 4 is a block diagram of a software architecture 400 applied to each processor 304-306 according to an embodiment of the present invention. The software architecture 400 comprises at least one task. In the present example, however, a number of tasks 402A-402N are shown in FIG. 4. Each task 402 comprises at least one queue 404 for processing at least one service application. In the present example, a number of queues 404A-404N are shown. The second processor 306 can have any number and combination of tasks 402 and corresponding service queues 404." Khawand '528 at col. 3, ll. 52–61. "During operation, the queue 404A begins to fill with service applications received by the second processor 306. This process is represented as a capacity level 503A. As the capacity level 503A traversing the MID 506 threshold, a new capacity level 503B is created. The new capacity level 503B is artificially created so that in the process of reducing the capacity level represented by 504A (i.e., capacity level 503B in the reverse direction) more than one service application must be processed by the second processor 306 before reaching the MID 506 threshold. When reduction in the queue 404A reaches the MID 506 threshold, the capacity level 504A jumps to the capacity level 504B. From this point, increases in the size of the queue 404A again requires receiving more than one service application to reach the MID 506 threshold, thereby restarting the cycle described above." Khawand '528 at col. 4, ll. 37–51. "In light of the foregoing description, it should be recognized that embodiments in the present invention could be realized in hardware, software, or a combination of hardware and software. These embodiments could also be realized in numerous configurations contemplated to be within the scope and spirit of the claims below. It should also be understood that the claims are intended to cover the structures described herein as performing

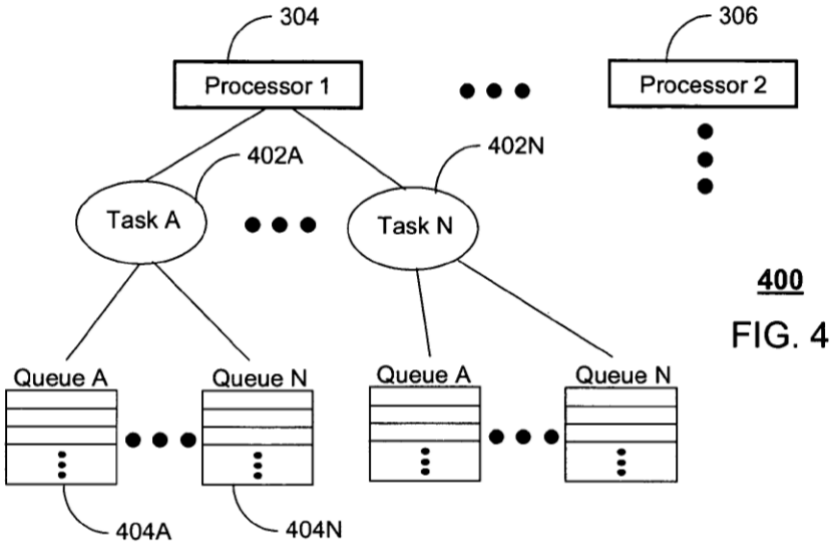
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	the recited function and not only structural equivalents. The claims are sufficiently general to include equivalent structures." Khawand '528 at col. 6, ll. 14–23. "For example, a software implementation of a method and a hardware implementation of the same method can not be structural equivalents in that the software implementation is dependent on a processing system for execution, while the hardware implementation may have self-contained processing means. It is well known the art, however, that software and hardware implementations may be designed to be equivalent structures generating the same results. Accordingly, all equivalent modifications and additions to the description above are intended to be inclusive of the scope the following claims." Khawand '528 at col. 6, ll. 24–34.
[1b] an output queue implemented in hardware;	Khawand '528 teaches "an output queue implemented in hardware," including at least the following exemplary citations. Plaintiffs have cited the use of an output queue implemented in memory through software as allegedly satisfying limitation [1b]. *See* Am. Compl. at ¶59; '596 Infringement Contentions at 58–72. To the extent that Plaintiffs argue that Khawand '528's output queue is in memory Khawand '528 teaches or renders obvious claim limitation [1b] under Plaintiffs' apparent interpretation of the claim. For example, Khawand '528 teaches an output queue implemented in hardware, by disclosing, as non–limiting examples, results from processing tasks being returned to the delegating processor. *See, e.g.:* Defendants have proposed a construction that "queue implemented in hardware" means "queue implemented in physical circuits without software intervention." The output queue in Khawand '528 meets Defendants' construction because, for example, Khawand '528 teaches a hardware implementation of its teachings, which includes its teachings of queues. *See, e.g.*, Khawand '528 at 6:14–17 ("In light of the foregoing description, it should be recognized that embodiments in the present invention could be realized in hardware, software, or a combination of hardware and software.").

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	  Khawand ’528 at Figures 2 and 3.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figure 4. “FIG. 3 shows a detailed view of the processing system 206, comprising a conventional memory 302, a plurality of processors 304-306 (shown by way of example as first and second processors) and an inter-processor manager 308 according to an embodiment of the present invention. The memory 302 can be a single or a combination of conventional memory devices (e.g., dynamic random access memory, flash memory, etc.) operated on individually or jointly by the elements 304-308 of the processing system 206. The plurality of processors 304-306 employ conventional processing elements such as, for example, an ARM processor. The inter-processor manager 308 can comprise a conventional application specific integrated circuit (ASIC), a microprocessor, a state machine, or other processing means that is capable of operating according to the method 600 of FIG. 6. Alternatively, the processing function of the inter-processor manager 308 can take the form of hardware, software or combinations thereof operating in whole or in part from the first or second processors 304, or independent hardware. The processing elements 302-308 of the processing system 206 are interconnected by a conventional communication bus 310. The construction of the bus 310 can be physical or logical. In the former, the physical connection can be a conventional serial or parallel bus for transmitting signals between devices sourced by a conventional transceiver (e.g., universal asynchronous receiver transmitter—UART, or universal serial bus—USB driver). Alternatively, a portion or whole of the communication bus 310 can represent a

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	logical connection such as software components sharing data structures. The processing elements 302-308 coupled to the communication bus 310 can be represented in a single integrated circuit (IC) device with sub-elements operating therein, or a combination of ICs, each operating as a corresponding one of the elements 302-308 of the processing system 206." Khawand '528 at col. 3, ll. 18–51. "In light of the foregoing description, it should be recognized that embodiments in the present invention could be realized in hardware, software, or a combination of hardware and software. These embodiments could also be realized in numerous configurations contemplated to be within the scope and spirit of the claims below. It should also be understood that the claims are intended to cover the structures described herein as performing the recited function and not only structural equivalents. The claims are sufficiently general to include equivalent structures." Khawand '528 at col. 6, ll. 14–23. "For example, a software implementation of a method and a hardware implementation of the same method can not be structural equivalents in that the software implementation is dependent on a processing system for execution, while the hardware implementation may have self-contained processing means. It is well known the art, however, that software and hardware implementations may be designed to be equivalent structures generating the same results. Accordingly, all equivalent modifications and additions to the description above are intended to be inclusive of the scope the following claims." Khawand '528 at col. 6, ll. 24–34.
[1c] a first processing element having access to said input queue and said output queue;	Khawand '528 teaches "a first processing element having access to said input queue and said output queue," including at least the following exemplary citations. For example, Khawand '528 teaches a first processing element having access to said input queue and said output queue, by disclosing, as non–limiting examples, a plurality of processors in a multiprocessor system. Any one of these processors satisfies the first processing element. Each processor has access to both the input queue and output queue. *See, e.g.:* "A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second

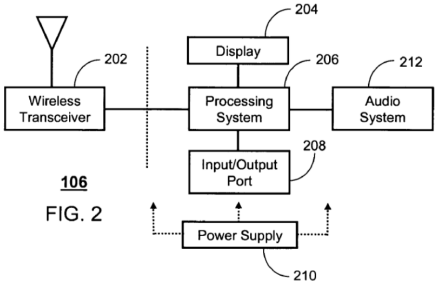
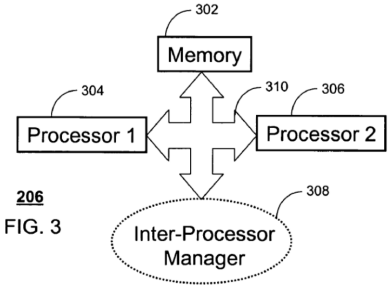
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at Abstract.   Khawand '528 at Figures 2 and 3.

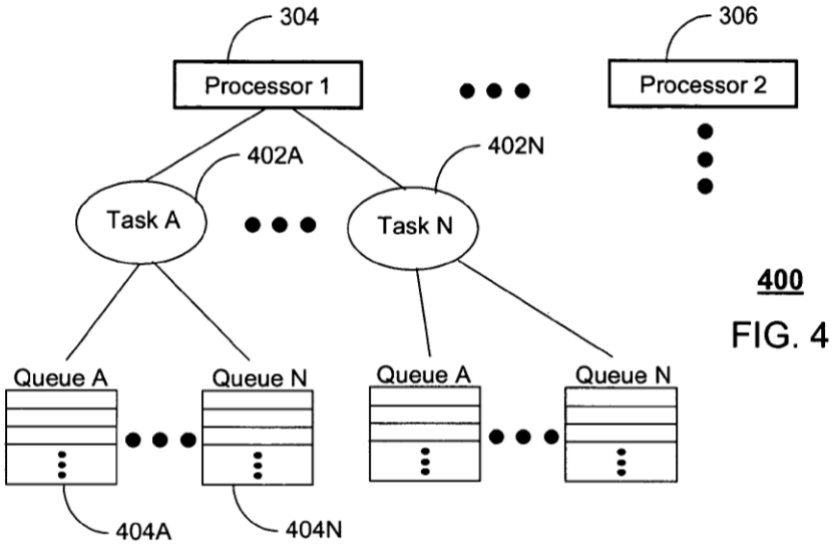
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figure 4. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 35–48. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 49–61.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	"In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "The processing system 206 is the processing center for the mobile device 106, which controls operations of the foregoing components 202-212. The processing system 206 controls singly or concurrently voice processing (e.g., voice calls), multi-media processing (video MPEG 4 player), data message processing (e.g., application downloads, short message system exchanges, etc.), and other house-keeping functions of the mobile device 106. The processing system 206 uses an operating system for distributing service applications in a manner consistent with method 600 of FIG. 6. Service applications can represent all inclusive software algorithms (e.g., an MPEG 4 video decoder), or portions thereof subdivided into tasks (e.g., input/output drivers, display drivers, etc.) and/or primitive instructions such as microcode." Khawand '528 at col. 3, ll. 4–17. "FIG. 3 shows a detailed view of the processing system 206, comprising a conventional memory 302, a plurality of processors 304-306 (shown by way of example as first and second processors) and an inter-processor manager 308 according to an embodiment of the present invention. The memory 302 can be a single or a combination of conventional memory devices (e.g., dynamic random access memory, flash memory, etc.) operated on individually or jointly by the elements 304-308 of the processing system 206. The plurality of processors 304-306 employ conventional processing elements such as, for example, an ARM processor. The inter-processor manager 308 can comprise a conventional application specific integrated circuit (ASIC), a microprocessor, a state machine, or other processing means that is capable of operating according to the method 600 of FIG. 6. Alternatively, the processing function of the inter-processor manager 308 can take the form of hardware, software or combinations thereof operating in whole or in part from the first or second processors 304, or independent hardware. The processing elements 302-308 of the processing system 206 are interconnected by a conventional communication bus 310. The construction of the bus 310 can be physical or logical. In the former, the

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	physical connection can be a conventional serial or parallel bus for transmitting signals between devices sourced by a conventional transceiver (e.g., universal asynchronous receiver transmitter—UART, or universal serial bus—USB driver). Alternatively, a portion or whole of the communication bus 310 can represent a logical connection such as software components sharing data structures. The processing elements 302-308 coupled to the communication bus 310 can be represented in a single integrated circuit (IC) device with sub-elements operating therein, or a combination of ICs, each operating as a corresponding one of the elements 302-308 of the processing system 206." Khawand '528 at col. 3, ll. 18–51. "FIG. 4 is a block diagram of a software architecture 400 applied to each processor 304-306 according to an embodiment of the present invention. The software architecture 400 comprises at least one task. In the present example, however, a number of tasks 402A-402N are shown in FIG. 4. Each task 402 comprises at least one queue 404 for processing at least one service application. In the present example, a number of queues 404A-404N are shown. The second processor 306 can have any number and combination of tasks 402 and corresponding service queues 404." Khawand '528 at col. 3, ll. 52–61. "FIG. 6 is a flow chart depicting a method 600 for substantially improving the processing efficiency of service applications operating in the mobile device 106 according to an embodiment of the present invention. We begin with step 602, where the first and second processors 304-306 of the mobile device 106 are programmed to delegate a service application from the first processor 304 to a queue 404 of the second processor 306. The rationale for this delegation can be, for instance, the first processor 304 running out of processing capacity for a particular task 402 or sub-task (represented by a particular queue 404)." Khawand '528 at col. 3, l. 62 – col. 4, l. 5. "During operation, the queue 404A begins to fill with service applications received by the second processor 306. This process is represented as a capacity level 503A. As the capacity level 503A traversing the MID 506 threshold, a new capacity level 503B is created. The new capacity level 503B is artificially created so that in the process of reducing the capacity level represented by 504A (i.e., capacity level 503B in the reverse direction) more than one service application must be processed by the second processor 306 before reaching the MID 506 threshold. When reduction in the queue 404A reaches the MID 506 threshold, the capacity level 504A jumps to the capacity level 504B. From this point, increases in the size of the queue 404A again requires receiving more than one service application to reach the MID 506 threshold, thereby restarting the cycle described above." Khawand '528 at col. 4, ll. 37–51. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49.
[1d] at least one second processing element in communication with said first processing element;	Khawand '528 teaches "at least one second processing element in communication with said first processing element," including at least the following exemplary citations. For example, Khawand '528 teaches at least one second processing element in communication with said first processing element, by disclosing, as non–limiting examples, an inter–processor manager. The inter–processor manager, the second processing element is in communication with the first processing element. *See, e.g.:* "A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at Abstract.

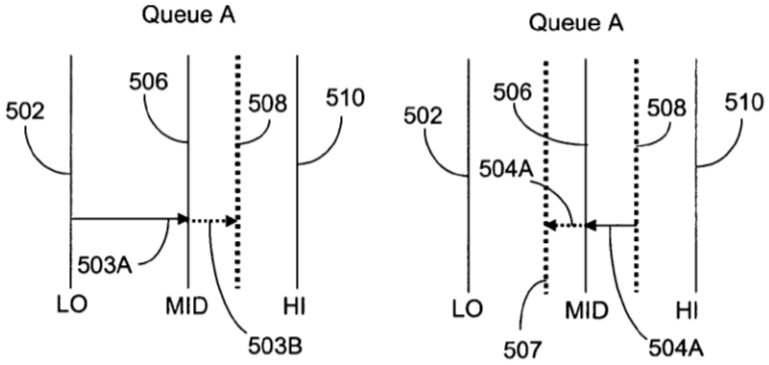
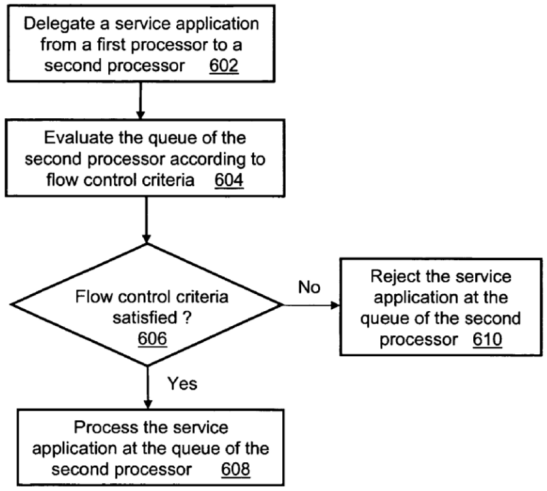
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	  Khawand ’528 at Figures 2 and 3.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figure 4. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 35–48. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 49–61.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	"In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "The processing system 206 is the processing center for the mobile device 106, which controls operations of the foregoing components 202-212. The processing system 206 controls singly or concurrently voice processing (e.g., voice calls), multi-media processing (video MPEG 4 player), data message processing (e.g., application downloads, short message system exchanges, etc.), and other house-keeping functions of the mobile device 106. The processing system 206 uses an operating system for distributing service applications in a manner consistent with method 600 of FIG. 6. Service applications can represent all inclusive software algorithms (e.g., an MPEG 4 video decoder), or portions thereof subdivided into tasks (e.g., input/output drivers, display drivers, etc.) and/or primitive instructions such as microcode." Khawand '528 at col. 3, ll. 4–17. "FIG. 3 shows a detailed view of the processing system 206, comprising a conventional memory 302, a plurality of processors 304-306 (shown by way of example as first and second processors) and an inter-processor manager 308 according to an embodiment of the present invention. The memory 302 can be a single or a combination of conventional memory devices (e.g., dynamic random access memory, flash memory, etc.) operated on individually or jointly by the elements 304-308 of the processing system 206. The plurality of processors 304-306 employ conventional processing elements such as, for example, an ARM processor. The inter-processor manager 308 can comprise a conventional application specific integrated circuit (ASIC), a microprocessor, a state machine, or other processing means that is capable of operating according to the method 600 of FIG. 6. Alternatively, the processing function of the inter-processor manager 308 can take the form of hardware, software or combinations thereof operating in whole or in part from the first or second processors 304, or independent hardware. The processing elements 302-308 of the processing system 206 are interconnected by a conventional communication bus 310. The construction of the bus 310 can be physical or logical. In the former, the

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	physical connection can be a conventional serial or parallel bus for transmitting signals between devices sourced by a conventional transceiver (e.g., universal asynchronous receiver transmitter—UART, or universal serial bus—USB driver). Alternatively, a portion or whole of the communication bus 310 can represent a logical connection such as software components sharing data structures. The processing elements 302-308 coupled to the communication bus 310 can be represented in a single integrated circuit (IC) device with sub-elements operating therein, or a combination of ICs, each operating as a corresponding one of the elements 302-308 of the processing system 206." Khawand '528 at col. 3, ll. 18–51. "FIG. 6 is a flow chart depicting a method 600 for substantially improving the processing efficiency of service applications operating in the mobile device 106 according to an embodiment of the present invention. We begin with step 602, where the first and second processors 304-306 of the mobile device 106 are programmed to delegate a service application from the first processor 304 to a queue 404 of the second processor 306. The rationale for this delegation can be, for instance, the first processor 304 running out of processing capacity for a particular task 402 or sub-task (represented by a particular queue 404)." Khawand '528 at col. 3, l. 62 – col. 4, l. 5. "During operation, the queue 404A begins to fill with service applications received by the second processor 306. This process is represented as a capacity level 503A. As the capacity level 503A traversing the MID 506 threshold, a new capacity level 503B is created. The new capacity level 503B is artificially created so that in the process of reducing the capacity level represented by 504A (i.e., capacity level 503B in the reverse direction) more than one service application must be processed by the second processor 306 before reaching the MID 506 threshold. When reduction in the queue 404A reaches the MID 506 threshold, the capacity level 504A jumps to the capacity level 504B. From this point, increases in the size of the queue 404A again requires receiving more than one service application to reach the MID 506 threshold, thereby restarting the cycle described above." Khawand '528 at col. 4, ll. 37–51. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49.

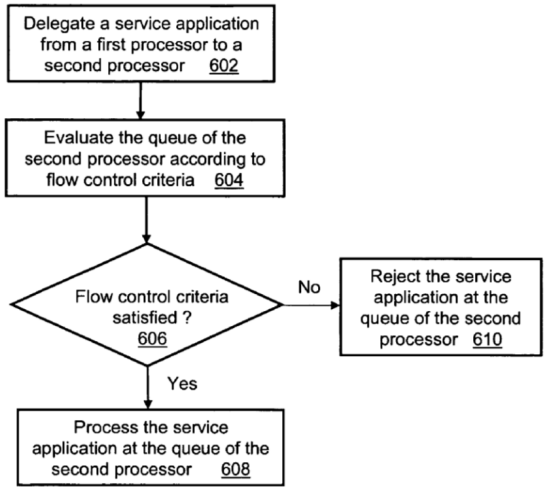
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
[1e] a reservation register implemented in hardware storing a value indicative of available space in said input queue, said reservation register accessible by both said first processing element and said at least one second processing element;	Khawand ’528 teaches “a reservation register implemented in hardware storing a value indicative of available space in said input queue, said reservation register accessible by both said first processing element and said at least one second processing element,” including at least the following exemplary citations. Plaintiffs have cited the use of data structures in memory storing a value indicative of available space, e.g., qstate, as allegedly satisfying limitation [1e]. *See* Am. Compl. at ¶62; ’596 Infringement Contentions at 100–137. To the extent that Plaintiffs argue that Khawand ’528’s queue state is in memory, Khawand ’528 discloses or renders obvious claim limitation [1e] under Plaintiffs’ apparent interpretation of the claim. For example, Khawand ’528 teaches a reservation register implemented in hardware storing a value indicative of available space in said input queue, said reservation register accessible by both said first processing element and said at least one second processing element, by disclosing, as non–limiting examples, queue state that is checked to see if the queue satisfies flow control criteria. The queue state stores a value of indicative space and is accessible by both the first and second processing elements discussed above. *See, e.g.:* “A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at Abstract.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 FIG. 5 **500** Khawand ’528 at Figure 5.  FIG. 6 **600** Khawand ’528 at Figure 6.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	"In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 35–48. "In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "In step 604, the queue 404 of the second processor 306 is evaluated according to flow control criteria. In a first embodiment, the evaluation step can take place at the inter-processor manager 308. Alternatively, the evaluation step can take place at the second processor 306. It will be appreciated by one of ordinary skill in the art that the method 600 described herein can be distributed to the processors 304-306, centralized in the inter-processor manager 308, or any combinations thereof.

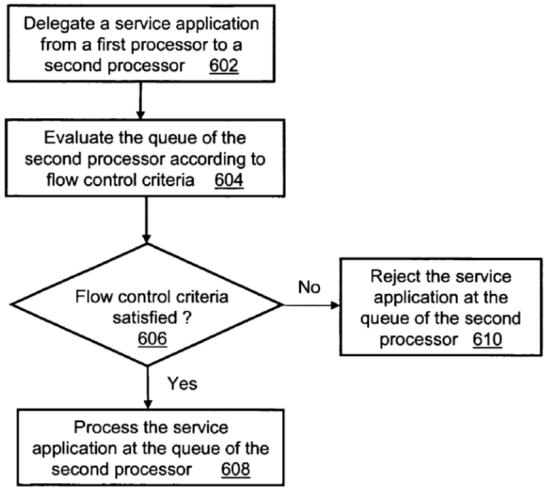
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	Returning to step 604, the flow control criteria can include a capacity threshold for each queue 404. The capacity threshold indicates the capacity of the queue 404 for processing service applications. In one embodiment, the service application is processed in step 608 by the second processor 306 if at decision block 606 it is determined that the capacity threshold of the queue 404 is not exceeded (i.e., the flow control criteria has been satisfied). On the other hand, the service application is rejected by the second processor 306 in step 610 upon determining at decision block 606 that the capacity threshold of the queue 404 has been exceeded. This embodiment reflects a fixed capacity threshold for managing the capacity of the queues 404." Khawand '528 at col. 4, ll. 6–27. "Alternatively, the capacity threshold can be variable. In such an embodiment, the capacity threshold can follow a hysteresis effect as demonstrated by the graph 500 of FIG. 5. In FIG. 5, the queue 404A (of FIG. 4) has three thresholds LO 502, MID 506 and HI 508. The LO 502 threshold represents an empty or nearly empty queue 502. The MID 506 threshold represents a half-full or nearly half-full queue 402A. And the HI 508 threshold represents a full or nearly full queue 404A." Khawand '528 at col. 4, ll. 28–35. "Reestablishing reception of service applications at the corresponding one of the group of processing elements mentioned above occurs upon satisfying the flow control criteria as described above. When this happens, the second processor 306 submits a resume command to the inter-processor manager 308 indicating the corresponding one of the group of processing elements is available to process service applications, which in turn synchronizes the first processor 304 according to the resume command. In an alternative embodiment where the inter-processor manager 308 is not used, the suspend and resume commands are transmitted from the second processor 306 to the first processor 304." Khawand '528 at col. 5, ll. 18–29. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49. "In light of the foregoing description, it should be recognized that embodiments in the present invention could be realized in hardware, software, or a combination of hardware and software. These embodiments could also be realized in numerous configurations contemplated to be within the scope and spirit of the claims below. It should also be understood that the claims are intended to cover the structures described herein as performing the recited function and not only structural equivalents. The claims are sufficiently general to include equivalent structures." Khawand '528 at col. 6, ll. 14–23.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	"For example, a software implementation of a method and a hardware implementation of the same method can not be structural equivalents in that the software implementation is dependent on a processing system for execution, while the hardware implementation may have self-contained processing means. It is well known the art, however, that software and hardware implementations may be designed to be equivalent structures generating the same results. Accordingly, all equivalent modifications and additions to the description above are intended to be inclusive of the scope the following claims." Khawand '528 at col. 6, ll. 24–34. "5. The method of claim 1, further comprising the step of requesting at the first processor an update from the queue of the second processor prior to delegating the service application thereto." Khawand '528 at col. 7, ll. 28–31.
[1f] computer storage storing instructions, which when executed by said at least one second processing element:	Khawand '528 teaches computer storage storing instructions executed by said at least one second processing element, including at least the following exemplary citations. For example, Khawand '528 teaches computer storage storing instructions that are executed by said at least one second processing element, by disclosing, as non–limiting examples, a processor that implements the inter–processor manager functionality by executing computer instructions from computer storage. The inter–processor manager executes those instructions stored in the computer storage. *See, e.g.:* "A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at Abstract.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 FIG. 6 600 Khawand ’528 at Figure 6. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 49–61. “In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "FIG. 3 shows a detailed view of the processing system 206, comprising a conventional memory 302, a plurality of processors 304-306 (shown by way of example as first and second processors) and an inter-processor manager 308 according to an embodiment of the present invention. The memory 302 can be a single or a combination of conventional memory devices (e.g., dynamic random access memory, flash memory, etc.) operated on individually or jointly by the elements 304-308 of the processing system 206. The plurality of processors 304-306 employ conventional processing elements such as, for example, an ARM processor. The inter-processor manager 308 can comprise a conventional application specific integrated circuit (ASIC), a microprocessor, a state machine, or other processing means that is capable of operating according to the method 600 of FIG. 6. Alternatively, the processing function of the inter-processor manager 308 can take the form of hardware, software or combinations thereof operating in whole or in part from the first or second processors 304, or independent hardware. The processing elements 302-308 of the processing system 206 are interconnected by a conventional communication bus 310. The construction of the bus 310 can be physical or logical. In the former, the physical connection can be a conventional serial or parallel bus for transmitting signals between devices sourced by a conventional transceiver (e.g., universal asynchronous receiver transmitter—UART, or universal serial bus—USB driver). Alternatively, a portion or whole of the communication bus 310 can represent a logical connection such as software components sharing data structures. The processing elements 302-308 coupled to the communication bus 310 can be represented in a single integrated circuit (IC) device with sub-elements operating therein, or a combination of ICs, each operating as a corresponding one of the elements 302-308 of the processing system 206." Khawand '528 at col. 3, ll. 18–51. "In step 604, the queue 404 of the second processor 306 is evaluated according to flow control criteria. In a first embodiment, the evaluation step can take place at the inter-processor manager 308. Alternatively, the evaluation step can take place at the second processor 306. It will be appreciated by one of ordinary skill in the art that the method 600 described herein can be distributed to the processors 304-306, centralized in the inter-processor manager 308, or any combinations thereof. Returning to step 604, the flow control criteria can include a capacity threshold for each queue 404. The capacity threshold indicates the capacity of the queue 404 for processing service applications. In one embodiment, the service application is processed in step 608 by the second processor 306 if at decision block 606 it is determined that the capacity threshold of the queue 404 is not exceeded (i.e., the flow control criteria

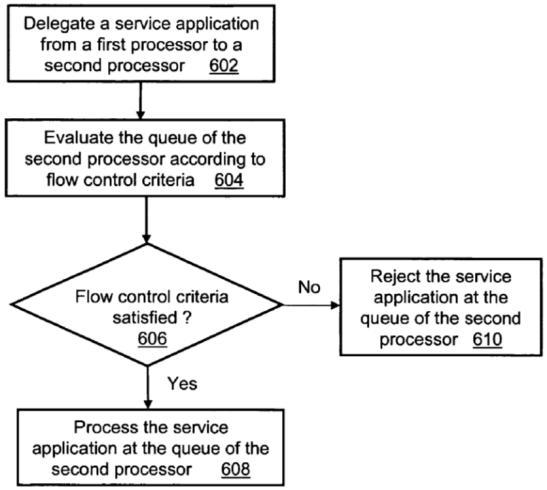
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	has been satisfied). On the other hand, the service application is rejected by the second processor 306 in step 610 upon determining at decision block 606 that the capacity threshold of the queue 404 has been exceeded. This embodiment reflects a fixed capacity threshold for managing the capacity of the queues 404." Khawand '528 at col. 4, ll. 6–27. "Reestablishing reception of service applications at the corresponding one of the group of processing elements mentioned above occurs upon satisfying the flow control criteria as described above. When this happens, the second processor 306 submits a resume command to the inter-processor manager 308 indicating the corresponding one of the group of processing elements is available to process service applications, which in turn synchronizes the first processor 304 according to the resume command. In an alternative embodiment where the inter-processor manager 308 is not used, the suspend and resume commands are transmitted from the second processor 306 to the first processor 304." Khawand '528 at col. 5, ll. 18–29. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49. "In light of the foregoing description, it should be recognized that embodiments in the present invention could be realized in hardware, software, or a combination of hardware and software. These embodiments could also be realized in numerous configurations contemplated to be within the scope and spirit of the claims below. It should also be understood that the claims are intended to cover the structures described herein as performing the recited function and not only structural equivalents. The claims are sufficiently general to include equivalent structures." Khawand '528 at col. 6, ll. 14–23. "5. The method of claim 1, further comprising the step of requesting at the first processor an update from the queue of the second processor prior to delegating the service application thereto." Khawand '528 at col. 7, ll. 28–31.
[1g] accesses said reservation register and reads said stored value;	Khawand '528 teaches "accesses said reservation register and reads said stored value," including at least the following exemplary citations. For example, Khawand '528 teaches that the second processing element executes instructions that causes it to access said reservation register and reads said stored value, by disclosing, as non–limiting examples,

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	evaluating the flow control criteria against a capacity threshold based on the queue statue. The inter–processor manager accesses and reads the stored value in the reservation register. *See, e.g.:* "A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at Abstract.  FIG. 6 600 Khawand '528 at Figure 6. "In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 35–48. "In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "In step 604, the queue 404 of the second processor 306 is evaluated according to flow control criteria. In a first embodiment, the evaluation step can take place at the inter-processor manager 308. Alternatively, the evaluation step can take place at the second processor 306. It will be appreciated by one of ordinary skill in the art that the method 600 described herein can be distributed to the processors 304-306, centralized in the inter-processor manager 308, or any combinations thereof. Returning to step 604, the flow control criteria can include a capacity threshold for each queue 404. The capacity threshold indicates the capacity of the queue 404 for processing service applications. In one embodiment, the service application is processed in step 608 by the second processor 306 if at decision block 606 it is determined that the capacity threshold of the queue 404 is not exceeded (i.e., the flow control criteria has been satisfied). On the other hand, the service application is rejected by the second processor 306 in step

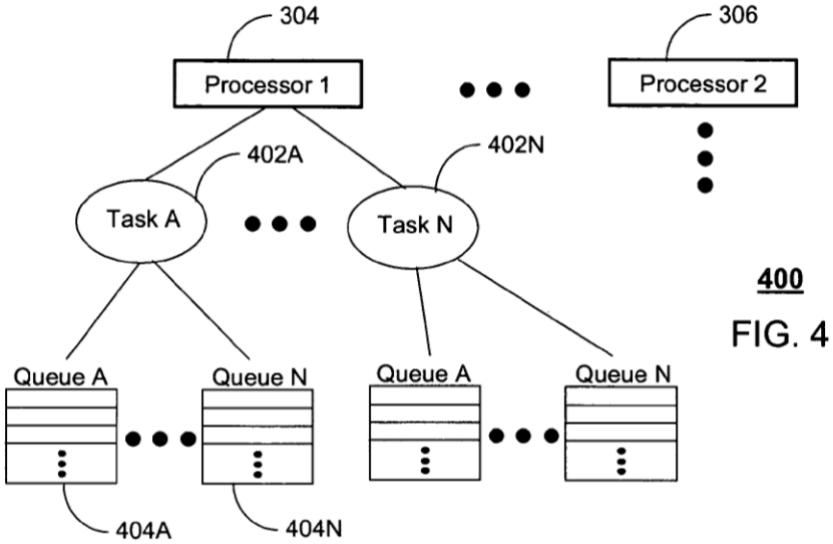
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	610 upon determining at decision block 606 that the capacity threshold of the queue 404 has been exceeded. This embodiment reflects a fixed capacity threshold for managing the capacity of the queues 404." Khawand '528 at col. 4, ll. 6–27. "Alternatively, the capacity threshold can be variable. In such an embodiment, the capacity threshold can follow a hysteresis effect as demonstrated by the graph 500 of FIG. 5. In FIG. 5, the queue 404A (of FIG. 4) has three thresholds LO 502, MID 506 and HI 508. The LO 502 threshold represents an empty or nearly empty queue 502. The MID 506 threshold represents a half-full or nearly half-full queue 402A. And the HI 508 threshold represents a full or nearly full queue 404A." Khawand '528 at col. 4, ll. 28–35. "Reestablishing reception of service applications at the corresponding one of the group of processing elements mentioned above occurs upon satisfying the flow control criteria as described above. When this happens, the second processor 306 submits a resume command to the inter-processor manager 308 indicating the corresponding one of the group of processing elements is available to process service applications, which in turn synchronizes the first processor 304 according to the resume command. In an alternative embodiment where the inter-processor manager 308 is not used, the suspend and resume commands are transmitted from the second processor 306 to the first processor 304." Khawand '528 at col. 5, ll. 18–29. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49. "In light of the foregoing description, it should be recognized that embodiments in the present invention could be realized in hardware, software, or a combination of hardware and software. These embodiments could also be realized in numerous configurations contemplated to be within the scope and spirit of the claims below. It should also be understood that the claims are intended to cover the structures described herein as performing the recited function and not only structural equivalents. The claims are sufficiently general to include equivalent structures." Khawand '528 at col. 6, ll. 14–23. "5. The method of claim 1, further comprising the step of requesting at the first processor an update from the queue of the second processor prior to delegating the service application thereto." Khawand '528 at col. 7, ll. 28–31.

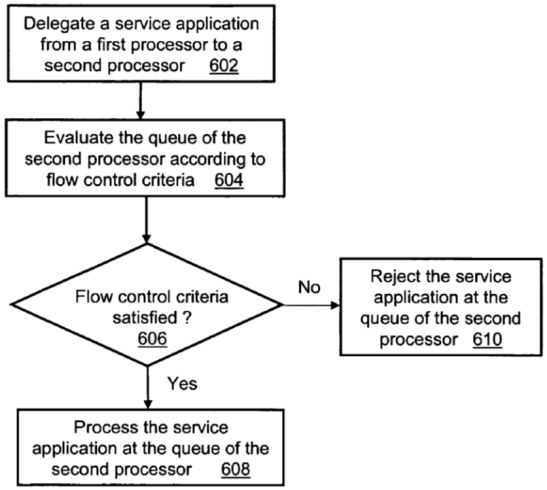
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
[1h] determines when said read value indicates available space in said input queue for said first processing element to issue a command;	Khawand ’528 teaches “determines when said read value indicates available space in said input queue for said first processing element to issue a command,” including at least the following exemplary citations. For example, Khawand ’528 teaches that the second processing element executes instructions that causes it to determine when said read value indicates available space in said input queue for said first processing element to issue a command, by disclosing, as non–limiting examples, evaluating the flow control criteria against a capacity threshold based on the queue statue. The inter–processor manager determines if the stored value indicates available space in the input queue for the first processing element to issue a command. *See, e.g.:* “A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at Abstract.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 FIG. 6 600 Khawand ’528 at Figure 6. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 35–48. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "In step 604, the queue 404 of the second processor 306 is evaluated according to flow control criteria. In a first embodiment, the evaluation step can take place at the inter-processor manager 308. Alternatively, the evaluation step can take place at the second processor 306. It will be appreciated by one of ordinary skill in the art that the method 600 described herein can be distributed to the processors 304-306, centralized in the inter-processor manager 308, or any combinations thereof. Returning to step 604, the flow control criteria can include a capacity threshold for each queue 404. The capacity threshold indicates the capacity of the queue 404 for processing service applications. In one embodiment, the service application is processed in step 608 by the second processor 306 if at decision block 606 it is determined that the capacity threshold of the queue 404 is not exceeded (i.e., the flow control criteria has been satisfied). On the other hand, the service application is rejected by the second processor 306 in step 610 upon determining at decision block 606 that the capacity threshold of the queue 404 has been exceeded. This embodiment reflects a fixed capacity threshold for managing the capacity of the queues 404." Khawand '528 at col. 4, ll. 6–27. "Alternatively, the capacity threshold can be variable. In such an embodiment, the capacity threshold can follow a hysteresis effect as demonstrated by the graph 500 of FIG. 5. In FIG. 5, the queue 404A (of FIG. 4) has three thresholds LO 502, MID 506 and HI 508. The LO 502 threshold represents an empty or nearly empty queue 502. The MID 506 threshold represents a half-full or nearly half-full queue 402A. And the HI 508 threshold represents a full or nearly full queue 404A." Khawand '528 at col. 4, ll. 28–35. "Reestablishing reception of service applications at the corresponding one of the group of processing elements mentioned above occurs upon satisfying the flow control criteria as described above. When this

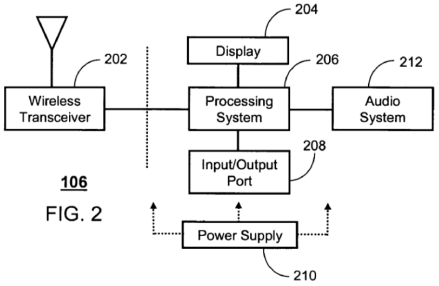
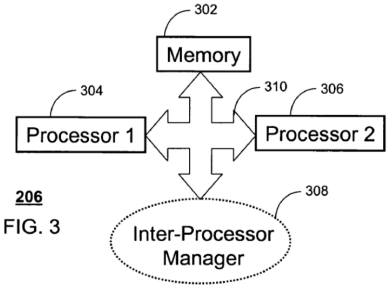
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	happens, the second processor 306 submits a resume command to the inter-processor manager 308 indicating the corresponding one of the group of processing elements is available to process service applications, which in turn synchronizes the first processor 304 according to the resume command. In an alternative embodiment where the inter-processor manager 308 is not used, the suspend and resume commands are transmitted from the second processor 306 to the first processor 304." Khawand '528 at col. 5, ll. 18–29. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49. "In light of the foregoing description, it should be recognized that embodiments in the present invention could be realized in hardware, software, or a combination of hardware and software. These embodiments could also be realized in numerous configurations contemplated to be within the scope and spirit of the claims below. It should also be understood that the claims are intended to cover the structures described herein as performing the recited function and not only structural equivalents. The claims are sufficiently general to include equivalent structures." Khawand '528 at col. 6, ll. 14–23. "5. The method of claim 1, further comprising the step of requesting at the first processor an update from the queue of the second processor prior to delegating the service application thereto." Khawand '528 at col. 7, ll. 28–31.
[1i] notifies said first processing element to issue said command to said input queue; and	Khawand '528 teaches "notifies said first processing element to issue said command to said input queue," including at least the following exemplary citations. For example, Khawand '528 teaches that the second processing element executes instructions that causes it to notify said first processing element to issue said command to said input queue, by disclosing, as non–limiting examples, issuing a resume command to the first processor when an input queue that was previously full has enough space for the command. The inter–processor manager notifies the first processing element to issue the command to the input queue. *See, e.g.:* "A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor

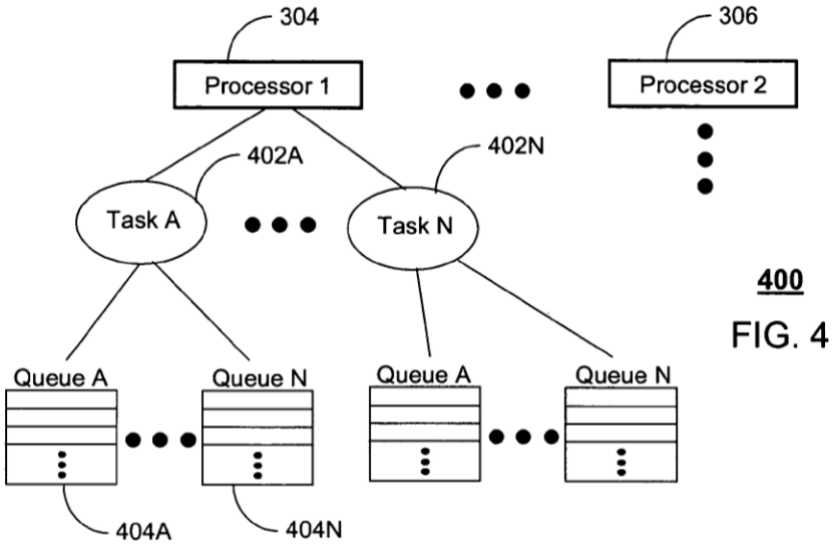
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at Abstract.  Khawand '528 at Figure 4.

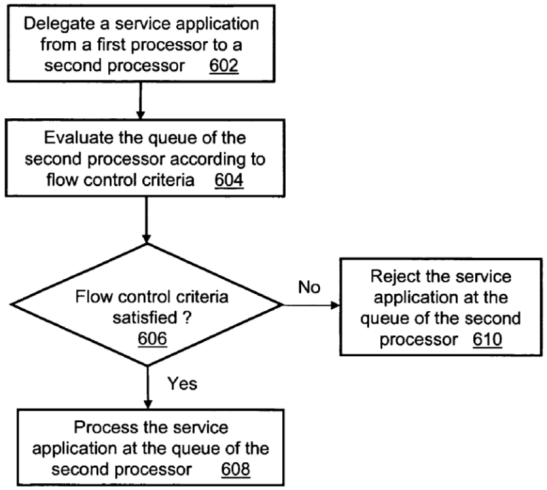
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 FIG. 6 600 Khawand ’528 at Figure 6. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 35–48. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "Reestablishing reception of service applications at the corresponding one of the group of processing elements mentioned above occurs upon satisfying the flow control criteria as described above. When this happens, the second processor 306 submits a resume command to the inter-processor manager 308 indicating the corresponding one of the group of processing elements is available to process service applications, which in turn synchronizes the first processor 304 according to the resume command. In an alternative embodiment where the inter-processor manager 308 is not used, the suspend and resume commands are transmitted from the second processor 306 to the first processor 304." Khawand '528 at col. 5, ll. 18–29. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49. "5. The method of claim 1, further comprising the step of requesting at the first processor an update from the queue of the second processor prior to delegating the service application thereto." Khawand '528 at col. 7, ll. 28–31.
[1j] wherein said first processing element receives notification from said at least one second processing element regarding issuing the	Khawand '528 teaches "wherein said first processing element receives notification from said at least one second processing element regarding issuing the command, issues said command to said input queue, and

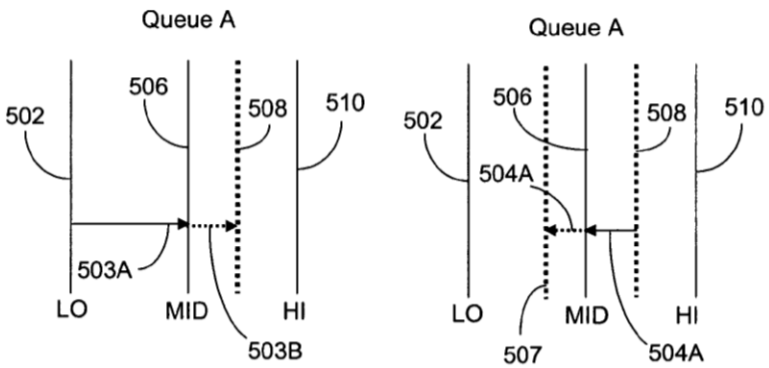
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
command, issues said command to said input queue, and receives a response corresponding to said command from said output queue.	receives a response corresponding to said command from said output queue," including at least the following exemplary citations. For example, Khawand '528 teaches wherein said first processing element receives notification from said at least one second processing element regarding issuing the command, issues said command to said input queue, and receives a response corresponding to said command from said output queue, by disclosing, as non–limiting examples, a first processor that is synchronize based on the resume notification, then issues the command, and receives a corresponding response from the output queue. *See, e.g.:* "A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at Abstract.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	  Khawand ’528 at Figures 2 and 3.

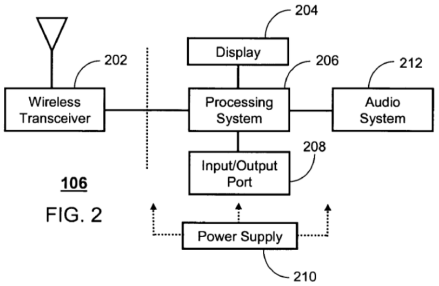
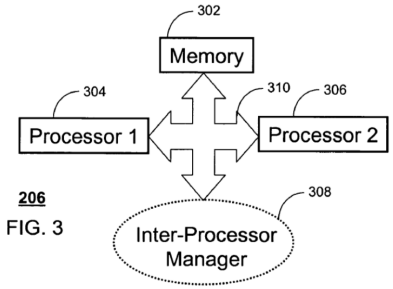
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figure 4.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 FIG. 6 600 Khawand ’528 at Figure 6. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 35–48. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and

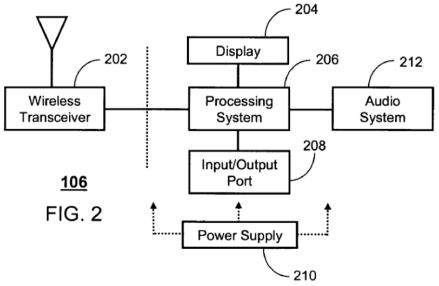
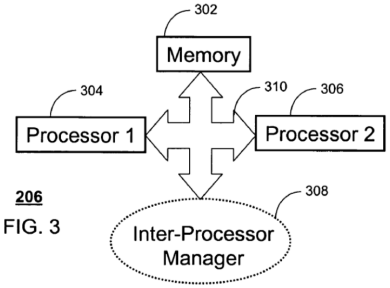
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "Reestablishing reception of service applications at the corresponding one of the group of processing elements mentioned above occurs upon satisfying the flow control criteria as described above. When this happens, the second processor 306 submits a resume command to the inter-processor manager 308 indicating the corresponding one of the group of processing elements is available to process service applications, which in turn synchronizes the first processor 304 according to the resume command. In an alternative embodiment where the inter-processor manager 308 is not used, the suspend and resume commands are transmitted from the second processor 306 to the first processor 304." Khawand '528 at col. 5, ll. 18–29. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49.
2. The system of claim 1, wherein said input queue is a first-in-first-out (FIFO) buffer.	
3. The system of claim 1, wherein said system is part of a storage area network (SAN) controller.	

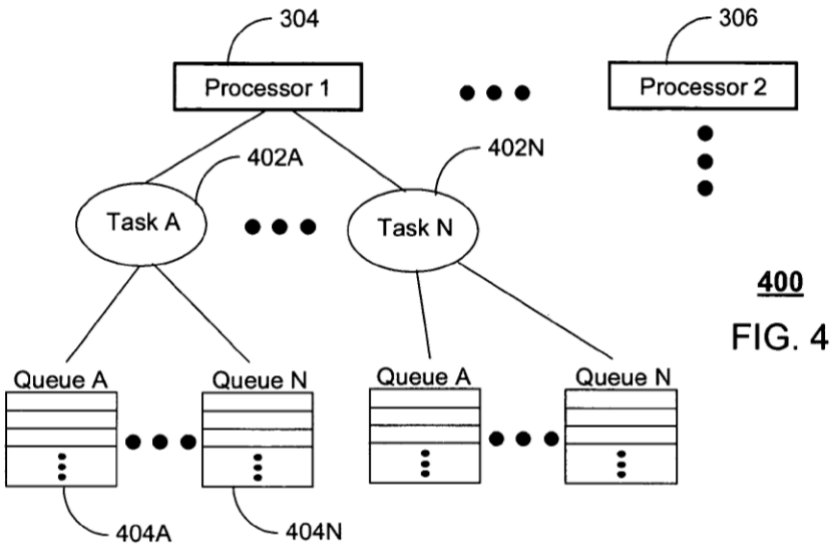
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
4. The system of claim 1, wherein said system is part of a network attached storage (NAS) controller.	
5. The system of claim 1, wherein said stored value is decremented by a predefined value prior to returning said stored value to said at least one second processing element.	Khawand '528 teaches "[t]he system of claim 1, wherein said stored value is decremented by a predefined value prior to returning said stored value to said at least one second processing element," including at least the following exemplary citations. For example, Khawand '528 teaches wherein said stored value is decremented by a predefined value prior to returning said stored value to said at least one second processing element by disclosing, as non–limiting examples, implementation of a hysteresis effect when managing the queues. The hysteresis effect decrements the stored value by a predefined value. *See, e.g.:*  FIG. 5 **500** Khawand '528 at Figure 5. "In step 604, the queue 404 of the second processor 306 is evaluated according to flow control criteria. In a first embodiment, the evaluation step can take place at the inter-processor manager 308. Alternatively, the evaluation step can take place at the second processor 306. It will be appreciated by one of ordinary skill in the art that the method 600 described herein can be distributed to the processors 304-306, centralized in the inter-processor manager 308, or any combinations thereof. Returning to step 604, the flow control criteria can include a capacity threshold for each queue 404. The capacity threshold indicates the capacity of the queue 404 for processing service applications. In one

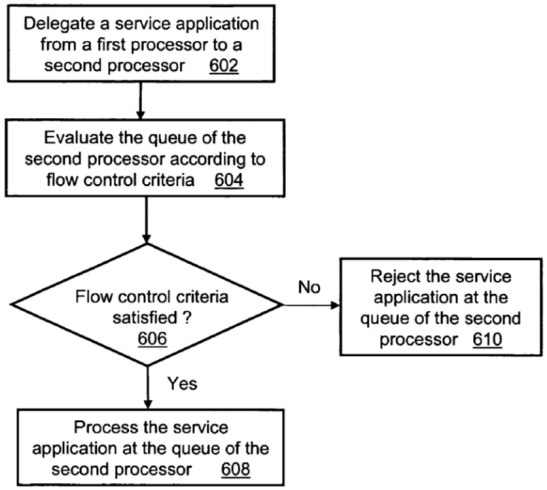
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	embodiment, the service application is processed in step 608 by the second processor 306 if at decision block 606 it is determined that the capacity threshold of the queue 404 is not exceeded (i.e., the flow control criteria has been satisfied). On the other hand, the service application is rejected by the second processor 306 in step 610 upon determining at decision block 606 that the capacity threshold of the queue 404 has been exceeded. This embodiment reflects a fixed capacity threshold for managing the capacity of the queues 404." Khawand '528 at col. 4, ll. 6–27. "Alternatively, the capacity threshold can be variable. In such an embodiment, the capacity threshold can follow a hysteresis effect as demonstrated by the graph 500 of FIG. 5. In FIG. 5, the queue 404A (of FIG. 4) has three thresholds LO 502, MID 506 and HI 508. The LO 502 threshold represents an empty or nearly empty queue 502. The MID 506 threshold represents a half-full or nearly half-full queue 402A. And the HI 508 threshold represents a full or nearly full queue 404A." Khawand '528 at col. 4, ll. 28–35. "During operation, the queue 404A begins to fill with service applications received by the second processor 306. This process is represented as a capacity level 503A. As the capacity level 503A traversing the MID 506 threshold, a new capacity level 503B is created. The new capacity level 503B is artificially created so that in the process of reducing the capacity level represented by 504A (i.e., capacity level 503B in the reverse direction) more than one service application must be processed by the second processor 306 before reaching the MID 506 threshold. When reduction in the queue 404A reaches the MID 506 threshold, the capacity level 504A jumps to the capacity level 504B. From this point, increases in the size of the queue 404A again requires receiving more than one service application to reach the MID 506 threshold, thereby restarting the cycle described above." Khawand '528 at col. 4, ll. 37–51. "The hysteresis effect demonstrated in FIG. 5 has the benefit of eliminating oscillations that can occur when increasing and decreasing the queue 404A around the MID 506 threshold. Using the MID 506 threshold as flow control criteria for managing the queues 404 of tasks 402 has the added benefit of a more stable environment for managing the queues 404 of each of the processors 304-306 by way of suspend and resume commands submitted in step 610." Khawand '528 at col. 4, ll. 52–59. "The variable capacity level utilizing a hysteresis behavior as described above provides a stable environment for managing queues 404 using suspend and resume commands. The system designer of the processing system 206 can select the MID 504 threshold and hysteresis range defined by boundaries 504-505 for each queue 404 in a manner that optimizes the overall performance of the system. Alternatively, this threshold and variable range can be dynamically altered by each of the queues 404 according to historical usage and

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	performance of service applications processed in each queue 404. This processing can also take place in the inter-processor manager 308 monitoring the plurality of processors." Khawand '528 at col. 5, ll. 30–41.
6. The system of claim 1, wherein at least one third processing element executes said command and notifies said output queue to issue said response.	Khawand '528 teaches "[t]he system of claim 1, wherein at least one third processing element executes said command and notifies said output queue to issue said response," including at least the following exemplary citations. For example, Khawand '528 teaches wherein at least one third processing element executes said command and notifies said output queue to issue said response, by disclosing, as non–limiting examples, a second processor that is not the first processor that executes the task delegated to it. The second processor executes the command and notifies the output queue to issue a response. *See, e.g.:*   Khawand '528 at Figures 2 and 3.
[7pre] A system for dividing and synchronizing a processing task	Khawand '528 teaches "[a] system for dividing and synchronizing a processing task across a plurality of processing elements," including at least the following exemplary citations.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
across a plurality of processing elements comprising:	*See* above at row [1pre].
[7a] an input queue implemented in hardware;	Khawand ’528 teaches “an input queue implemented in hardware,” including at least the following exemplary citations. *See* above at row [1a].
[7b] an output queue implemented in hardware;	Khawand ’528 teaches “an output queue implemented in hardware,” including at least the following exemplary citations. *See* above at row [1b].
[7c] a first processing element having access to said input queue and said output queue;	Khawand ’528 teaches “a first processing element having access to said input queue and said output queue,” including at least the following exemplary citations. *See* above at row [1c].
[7d] at least one second processing element in communication with said input queue;	Khawand ’528 teaches “at least one second processing element in communication with said input queue,” including at least the following exemplary citations. For example, Khawand ’528 teaches at least one second processing element in communication with said input queue, by disclosing, as non–limiting examples, an inter–processor manager that plays a centralized role in scheduling tasks. The inter–processor manager, the second processing element is in communication with the input queue. *See, e.g.:* “A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at Abstract.

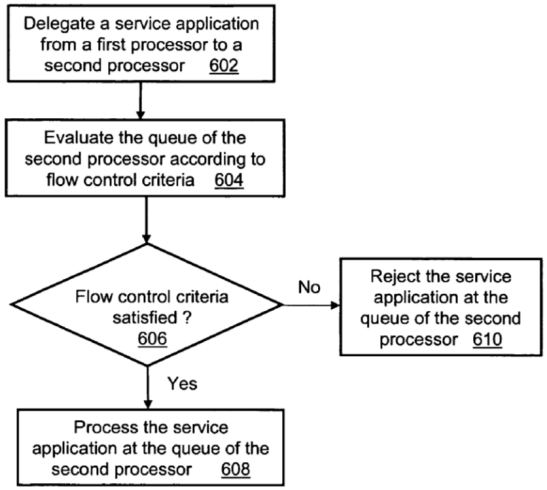
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	  Khawand ’528 at Figures 2 and 3.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figure 4.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 FIG. 6 600 Khawand ’528 at Figure 6. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 35–48. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "FIG. 3 shows a detailed view of the processing system 206, comprising a conventional memory 302, a plurality of processors 304-306 (shown by way of example as first and second processors) and an inter-processor manager 308 according to an embodiment of the present invention. The memory 302 can be a single or a combination of conventional memory devices (e.g., dynamic random access memory, flash memory, etc.) operated on individually or jointly by the elements 304-308 of the processing system 206. The plurality of processors 304-306 employ conventional processing elements such as, for example, an ARM processor. The inter-processor manager 308 can comprise a conventional application specific integrated circuit (ASIC), a microprocessor, a state machine, or other processing means that is capable of operating according to the method 600 of FIG. 6. Alternatively, the processing function of the inter-processor manager 308 can take the form of hardware, software or combinations thereof operating in whole or in part from the first or second processors 304, or independent hardware. The processing elements 302-308 of the processing system 206 are interconnected by a conventional communication bus 310. The construction of the bus 310 can be physical or logical. In the former, the physical connection can be a conventional serial or parallel bus for transmitting signals between devices sourced by a conventional transceiver (e.g., universal asynchronous receiver transmitter—UART, or universal serial bus—USB driver). Alternatively, a portion or whole of the communication bus 310 can represent a logical connection such as software components sharing data structures. The processing elements 302-308 coupled to the communication bus 310 can be represented in a single integrated circuit (IC) device with sub-elements operating therein, or a combination of ICs, each operating as a corresponding one of the elements 302-308 of the processing system 206." Khawand '528 at col. 3, ll. 18–51.

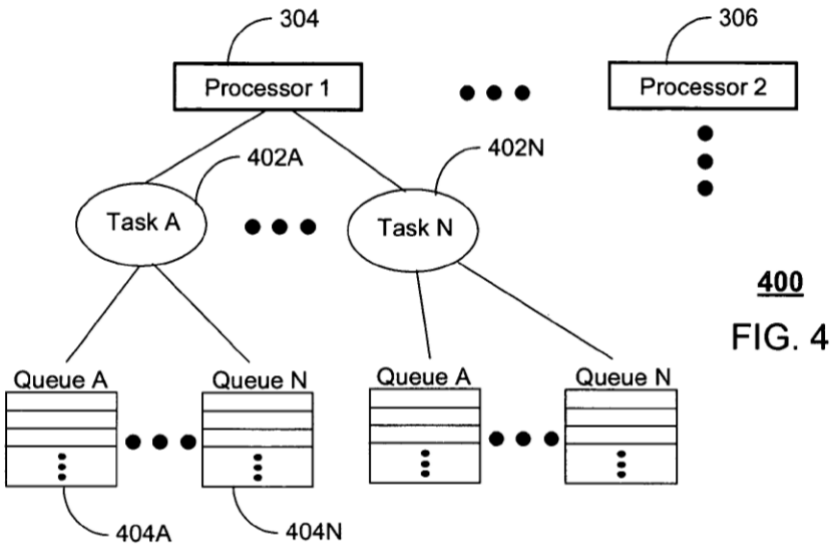
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	"FIG. 4 is a block diagram of a software architecture 400 applied to each processor 304-306 according to an embodiment of the present invention. The software architecture 400 comprises at least one task. In the present example, however, a number of tasks 402A-402N are shown in FIG. 4. Each task 402 comprises at least one queue 404 for processing at least one service application. In the present example, a number of queues 404A-404N are shown. The second processor 306 can have any number and combination of tasks 402 and corresponding service queues 404." Khawand '528 at col. 3, ll. 52–61. "FIG. 6 is a flow chart depicting a method 600 for substantially improving the processing efficiency of service applications operating in the mobile device 106 according to an embodiment of the present invention. We begin with step 602, where the first and second processors 304-306 of the mobile device 106 are programmed to delegate a service application from the first processor 304 to a queue 404 of the second processor 306. The rationale for this delegation can be, for instance, the first processor 304 running out of processing capacity for a particular task 402 or sub-task (represented by a particular queue 404)." Khawand '528 at col. 3, l. 62 – col. 4, l. 5.
[7e] a reservation register implemented in hardware storing a value indicative of available space in said input queue, said reservation register accessible by both said first processing element and said at least one second processing element;	Khawand '528 teaches "a reservation register implemented in hardware storing a value indicative of available space in said input queue, said reservation register accessible by both said first processing element and said at least one second processing element," including at least the following exemplary citations. *See* above at row [1e].
[7f] computer storage storing instructions, which when executed by said at least one second processing element:	Khawand '528 teaches computer storage storing instructions executed by said at least one second processing element, including at least the following exemplary citations. *See* above at row [1f].
[7g] receives a command from said first processing element;	Khawand '528 teaches "receives a command from said first processing element," including at least the following exemplary citations. For example, Khawand '528 teaches that the second processing element executes instructions that causes it to receive a command from said first processing element, by disclosing, as non–limiting examples, an inter–

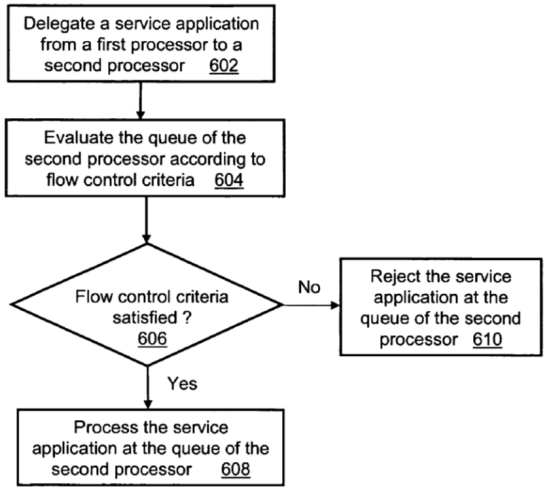
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	processor manager that plays a centralized role in scheduling tasks. The inter–processor manager, the second processing element receives a command from the first processing element. *See, e.g.:* "A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at Abstract.  FIG. 6 600 Khawand '528 at Figure 6. "In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 35–48. "In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "FIG. 6 is a flow chart depicting a method 600 for substantially improving the processing efficiency of service applications operating in the mobile device 106 according to an embodiment of the present invention. We begin with step 602, where the first and second processors 304-306 of the mobile device 106 are programmed to delegate a service application from the first processor 304 to a queue 404 of the second processor 306. The rationale for this delegation can be, for instance, the first processor 304 running out of processing capacity for a particular task 402 or sub-task (represented by a particular queue 404)." Khawand '528 at col. 3, l. 62 – col. 4, l. 5. "In step 604, the queue 404 of the second processor 306 is evaluated according to flow control criteria. In a first embodiment, the evaluation step can take place at the inter-processor manager 308. Alternatively, the evaluation step can take place at the second processor 306. It will be appreciated by one of ordinary skill in

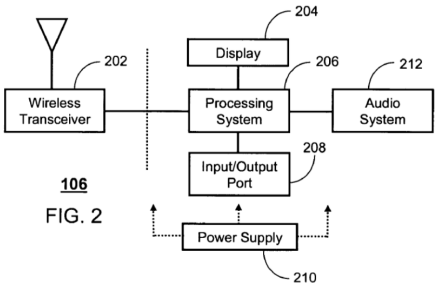
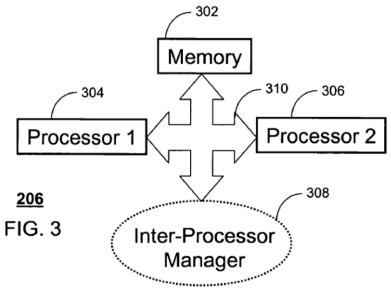
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	the art that the method 600 described herein can be distributed to the processors 304-306, centralized in the inter-processor manager 308, or any combinations thereof. Returning to step 604, the flow control criteria can include a capacity threshold for each queue 404. The capacity threshold indicates the capacity of the queue 404 for processing service applications. In one embodiment, the service application is processed in step 608 by the second processor 306 if at decision block 606 it is determined that the capacity threshold of the queue 404 is not exceeded (i.e., the flow control criteria has been satisfied). On the other hand, the service application is rejected by the second processor 306 in step 610 upon determining at decision block 606 that the capacity threshold of the queue 404 has been exceeded. This embodiment reflects a fixed capacity threshold for managing the capacity of the queues 404." Khawand '528 at col. 4, ll. 6–27. "During operation, the queue 404A begins to fill with service applications received by the second processor 306. This process is represented as a capacity level 503A. As the capacity level 503A traversing the MID 506 threshold, a new capacity level 503B is created. The new capacity level 503B is artificially created so that in the process of reducing the capacity level represented by 504A (i.e., capacity level 503B in the reverse direction) more than one service application must be processed by the second processor 306 before reaching the MID 506 threshold. When reduction in the queue 404A reaches the MID 506 threshold, the capacity level 504A jumps to the capacity level 504B. From this point, increases in the size of the queue 404A again requires receiving more than one service application to reach the MID 506 threshold, thereby restarting the cycle described above." Khawand '528 at col. 4, ll. 37–51. "Reestablishing reception of service applications at the corresponding one of the group of processing elements mentioned above occurs upon satisfying the flow control criteria as described above. When this happens, the second processor 306 submits a resume command to the inter-processor manager 308 indicating the corresponding one of the group of processing elements is available to process service applications, which in turn synchronizes the first processor 304 according to the resume command. In an alternative embodiment where the inter-processor manager 308 is not used, the suspend and resume commands are transmitted from the second processor 306 to the first processor 304." Khawand '528 at col. 5, ll. 18–29. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
[7h] accesses said reservation register and reads said stored value;	Khawand ’528 teaches “accesses said reservation register and reads said stored value,” including at least the following exemplary citations. *See* above at row [1g].
[7i] determines when said read value indicates available space in said input queue to issue said command;	Khawand ’528 teaches “determines when said read value indicates available space in said input queue to issue said command,” including at least the following exemplary citations. *See* above at row [1h].
[7j] issues said command to said input queue on behalf of said first processing element; and	Khawand ’528 teaches “issues said command to said input queue on behalf of said first processing element,” including at least the following exemplary citations. For example, Khawand ’528 teaches that the second processing element executes instructions that causes it to issue said command to said input queue on behalf of said first processing element, by disclosing, as non–limiting examples, an inter–processor manager that plays a centralized role in scheduling tasks. The inter–processor manager issues the command to the input queue on behalf of the first processing element. *See, e.g.:* “A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at Abstract.

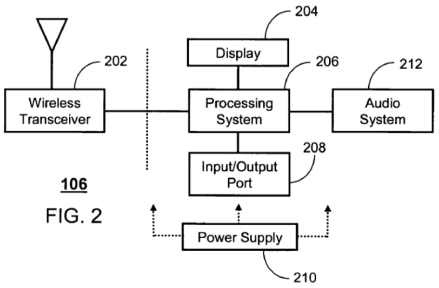
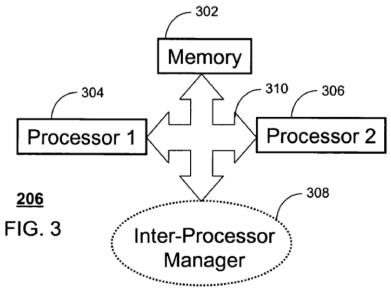
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figure 4.

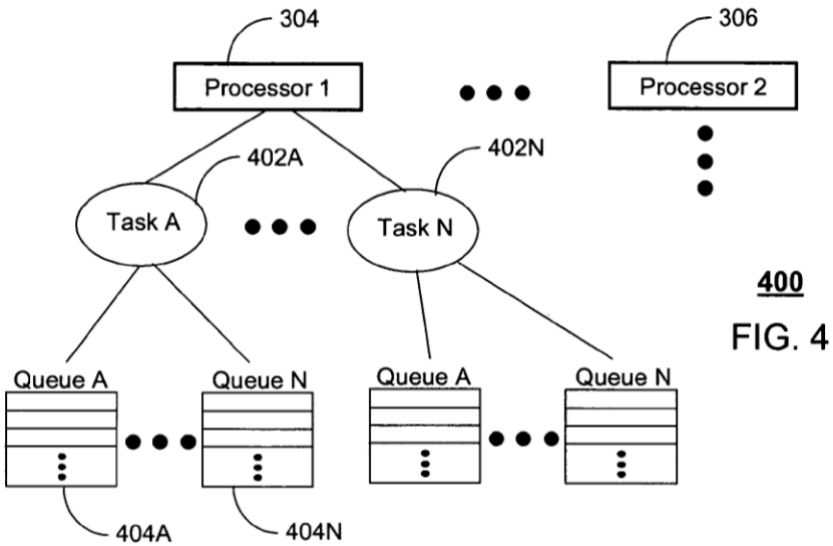
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 FIG. 6 600 Khawand ’528 at Figure 6. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 35–48. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and

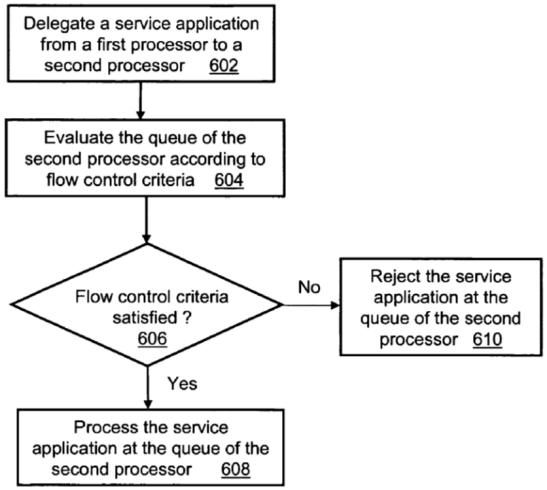
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "During operation, the queue 404A begins to fill with service applications received by the second processor 306. This process is represented as a capacity level 503A. As the capacity level 503A traversing the MID 506 threshold, a new capacity level 503B is created. The new capacity level 503B is artificially created so that in the process of reducing the capacity level represented by 504A (i.e., capacity level 503B in the reverse direction) more than one service application must be processed by the second processor 306 before reaching the MID 506 threshold. When reduction in the queue 404A reaches the MID 506 threshold, the capacity level 504A jumps to the capacity level 504B. From this point, increases in the size of the queue 404A again requires receiving more than one service application to reach the MID 506 threshold, thereby restarting the cycle described above." Khawand '528 at col. 4, ll. 37–51. "Referring back to step 602, the first processor 304 can be programmed to request an update regarding the capacity of the queue 404 of the second processor 306 before delegating the service application thereto. Alternatively, the first processor 304 can be programmed to request an update from a plurality of processors exceeding two, thereby selecting an optimal processor (e.g., the second processor 306) according to the status of the queue(s) 404 operating in said processor." Khawand '528 at col. 5, ll. 42–49.
[7k] wherein said first processing element receives a response corresponding to said command from said output queue.	Khawand '528 teaches "wherein said first processing element receives a response corresponding to said command from said output queue," including at least the following exemplary citations. For example, Khawand '528 teaches wherein said first processing element receives a response corresponding to said command from said output queue, by disclosing, as non–limiting examples, a first processor that

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	receives the results from the task it delegated to another processor. The first processor receives a response from the output queue corresponding to the command. *See, e.g.:*   Khawand ’528 at Figures 2 and 3.
8. The system of claim 7, wherein said input queue is a first-in-first-out (FIFO) buffer.	
9. The system of claim 7, wherein said system is part of a storage area network (SAN) controller.	
10. The system of claim 7, wherein said system is part of a network attached storage (NAS) controller.	

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
11. The system of claim 7, wherein said stored value is decremented by a predefined value prior to returning said stored value to said at least one second processing element.	Khawand '528 teaches "[t]he system of claim 7, wherein said stored value is decremented by a predefined value prior to returning said stored value to said at least one second processing element," including at least the following exemplary citations. *See* above at row 5.
12. The system of claim 7, wherein at least one third processing element executes said command and notifies said output queue to issue said response.	Khawand '528 teaches "[t]he system of claim 7, wherein at least one third processing element executes said command and notifies said output queue to issue said response," including at least the following exemplary citations. *See* above at row 6.
[13pre] A system for dividing and synchronizing a processing task across a plurality of processing elements comprising:	Khawand '528 teaches "[a] system for dividing and synchronizing a processing task across a plurality of processing elements," including at least the following exemplary citations. *See* above at row [1pre].
[13a] an input queue implemented in hardware;	Khawand '528 teaches "an input queue implemented in hardware," including at least the following exemplary citations. *See* above at row [1a].
[13b] an output queue implemented in hardware;	Khawand '528 teaches "an output queue implemented in hardware," including at least the following exemplary citations. *See* above at row [1b].
[13c] a first processing element having access to said input queue and said output queue;	Khawand '528 teaches "a first processing element having access to said input queue and said output queue," including at least the following exemplary citations. *See* above at row [1c].

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
[13d] at least one second processing element in communication with said input queue and said output queue;	Khawand ’528 teaches “at least one second processing element in communication with said input queue and said output queue,” including at least the following exemplary citations. For example, Khawand ’528 teaches at least one second processing element in communication with said input queue and said output queue, by disclosing, as non–limiting examples, an inter–processor manager that plays a centralized role in scheduling tasks. The inter–processor manager, the second processing element is in communication with the input and output queue. *See, e.g.:* “A multiprocessor system (206) having a plurality of processors (304-306), each processor capable of processing at least one queue (404A-404N) of at least one service application, and at least one task (402A-402N) comprising at least one of the at least one queue; and wherein a first processor and a second processor of the plurality of processors is each programmed to delegate (602) a service application from the first processor to a queue of the at least one queue of the second processor, evaluate (604-606) the queue at the second processor according to flow control criteria, process (608) the service application at the second processor upon satisfying the flow control criteria, and reject (610) the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at Abstract.   Khawand ’528 at Figures 2 and 3.

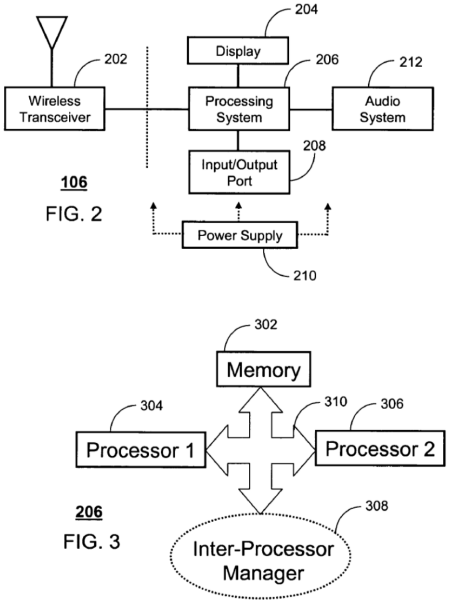
Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figure 4.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 FIG. 6 600 Khawand ’528 at Figure 6. “In a first embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. The multiprocessor system includes a method comprising the steps of delegating a service application from a first processor of the plurality of processors to a queue of the at least one queue of a second processor of the plurality of processors, evaluating the queue of the second processor according to flow control criteria, processing the service application at the second processor upon satisfying the flow control criteria, and rejecting the service application at the second processor upon failing to satisfy the flow control criteria.” Khawand ’528 at col. 1, ll. 35–48. “In a second embodiment of the present invention, a multiprocessor system has a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors is programmed to delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, ll. 49–61. "In a third embodiment of the present invention, a mobile device has a multiprocessor system including a plurality of processors, each processor capable of processing at least one queue of at least one service application, and at least one task comprising at least one of the at least one queue. A first processor and a second processor of the plurality of processors are programmed to receive and transmit messages by way of the transceiver to a communication system, delegate a service application from the first processor to a queue of the at least one queue of the second processor, evaluate the queue at the second processor according to flow control criteria, process the service application at the second processor upon satisfying the flow control criteria, and reject the service application at the second processor upon failing to satisfy the flow control criteria." Khawand '528 at col. 1, l. 62 – col. 2, l. 9. "FIG. 3 shows a detailed view of the processing system 206, comprising a conventional memory 302, a plurality of processors 304-306 (shown by way of example as first and second processors) and an inter-processor manager 308 according to an embodiment of the present invention. The memory 302 can be a single or a combination of conventional memory devices (e.g., dynamic random access memory, flash memory, etc.) operated on individually or jointly by the elements 304-308 of the processing system 206. The plurality of processors 304-306 employ conventional processing elements such as, for example, an ARM processor. The inter-processor manager 308 can comprise a conventional application specific integrated circuit (ASIC), a microprocessor, a state machine, or other processing means that is capable of operating according to the method 600 of FIG. 6. Alternatively, the processing function of the inter-processor manager 308 can take the form of hardware, software or combinations thereof operating in whole or in part from the first or second processors 304, or independent hardware. The processing elements 302-308 of the processing system 206 are interconnected by a conventional communication bus 310. The construction of the bus 310 can be physical or logical. In the former, the physical connection can be a conventional serial or parallel bus for transmitting signals between devices sourced by a conventional transceiver (e.g., universal asynchronous receiver transmitter—UART, or universal serial bus—USB driver). Alternatively, a portion or whole of the communication bus 310 can represent a logical connection such as software components sharing data structures. The processing elements 302-308 coupled to the communication bus 310 can be represented in a single integrated circuit (IC) device with sub-elements operating therein, or a combination of ICs, each operating as a corresponding one of the elements 302-308 of the processing system 206." Khawand '528 at col. 3, ll. 18–51.

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	"FIG. 4 is a block diagram of a software architecture 400 applied to each processor 304-306 according to an embodiment of the present invention. The software architecture 400 comprises at least one task. In the present example, however, a number of tasks 402A-402N are shown in FIG. 4. Each task 402 comprises at least one queue 404 for processing at least one service application. In the present example, a number of queues 404A-404N are shown. The second processor 306 can have any number and combination of tasks 402 and corresponding service queues 404." Khawand '528 at col. 3, ll. 52–61. "FIG. 6 is a flow chart depicting a method 600 for substantially improving the processing efficiency of service applications operating in the mobile device 106 according to an embodiment of the present invention. We begin with step 602, where the first and second processors 304-306 of the mobile device 106 are programmed to delegate a service application from the first processor 304 to a queue 404 of the second processor 306. The rationale for this delegation can be, for instance, the first processor 304 running out of processing capacity for a particular task 402 or sub-task (represented by a particular queue 404)." Khawand '528 at col. 3, l. 62 – col. 4, l. 5.
[13e] a reservation register implemented in hardware storing a value indicative of available space in said input queue, said reservation register accessible by both said first processing element and said at least one second processing element;	Khawand '528 teaches "a reservation register implemented in hardware storing a value indicative of available space in said input queue, said reservation register accessible by both said first processing element and said at least one second processing element," including at least the following exemplary citations. *See* above at row [1e].
[13f] computer storage storing instructions, which when executed by said at least one second processing element:	Khawand '528 teaches computer storage storing instructions that are executed by said at least one second processing element including at least the following exemplary citations. *See* above at row [1f].
[13g] receives a command from said first processing element;	Khawand '528 teaches "receives a command from said first processing element," including at least the following exemplary citations. *See* above at row [7g].

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
[13h] accesses said reservation register and reads said stored value;	Khawand ’528 teaches “accesses said reservation register and reads said stored value,” including at least the following exemplary citations. *See* above at row [1g].
[13i] determines when said read value indicates available space in said input queue to issue said command;	Khawand ’528 teaches “determines when said read value indicates available space in said input queue to issue said command,” including at least the following exemplary citations. *See* above at row [1h].
[13j] issues said command to said input queue on behalf of said first processing element; and	Khawand ’528 teaches “issues said command to said input queue on behalf of said first processing element,” including at least the following exemplary citations. *See* above at row [7j].
[13k] receives a response corresponding to said command from said output queue.	Khawand ’528 teaches “receives a response corresponding to said command from said output queue,” including at least the following exemplary citations. For example, Khawand ’528 teaches that the second processing element executes instructions that causes it to receive a response corresponding to said command from said output queue, by disclosing, as non–limiting examples, an inter–processor manager that plays a centralized role in scheduling tasks. The inter–processor manager receives a corresponding response from the output queue. *See, e.g.:*

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
	 Khawand ’528 at Figures 2 and 3.
14. The system of claim 13, wherein said input queue is a first-in-first-out (FIFO) buffer.	
15. The system of claim 13, wherein said system is either part of a storage area network (SAN) controller.	
16. The system of claim 13, wherein said system is part of a network attached storage (NAS) controller.	

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
17. The system of claim 13, wherein said stored value is decremented by a predefined value prior to returning said stored value to said at least one second processing element.	Khawand ’528 teaches “[t]he system of claim 13, wherein said stored value is decremented by a predefined value prior to returning said stored value to said at least one second processing element,” including at least the following exemplary citations. *See* above at row 5.
18. The system of claim 13, wherein at least one third processing element executes said command and notifies said output queue to issue said response.	Khawand ’528 teaches “[t]he system of claim 13, wherein at least one third processing element executes said command and notifies said output queue to issue said response,” including at least the following exemplary citations. *See* above at row 6.