

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

The asserted claims of U.S. Patent No. 10,873,753 are anticipated and/or rendered obvious in view of Cragon, “Memory Systems and Pipelined Processors” (“Cragon”) (avail. at PA032560-PA033769), alone or in combination with the knowledge of one of skill in the art or other prior art. Cragon was published and publicly available at least by 1996.

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 its 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 '753 patent is anticipated and/or rendered obvious in view of Cragon. 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
[1.pre] A platform for data flow processing comprising:	
[1.a] one or more swappable pods or cards in one or more chassis, coupled through a messaging interface network;	

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
[1.b] each of the one or more swappable pods or cards having one or more hardware modules or one or more software modules;	
[1.c] one or more of the plurality of swappable pods or cards having a portion for user-definable hardware modules or user-definable software modules;	
[1.d] the plurality of swappable pods or cards being user-configurable to implement data flow processing architectures; and	
[1.e] a network coupled to the one or more swappable pods or cards and supporting messaging-based communication using packets each having a header with a chassis identifier, a board identifier, a module identifier, an instance identifier, and a type	

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

Claim Language	Exemplary Evidence of Anticipation and/or Obviousness
identifier, so that each type of module, each instance of a type of module, and each module on each board in each chassis can be addressed through the header.	

2. The platform of claim 1, further comprising: at least a subset of the one or more swappable pods or cards having an option for direct connection to one or more pods, cards or components, bypassing the module messaging interface network.	
3. The platform of claim 1, wherein the one or more swappable pods or cards comprises at least one processor card having a at least one processor allocatable for offloading hardware tasks to software, orchestrating data flows, orchestrating	

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

configurations of hardware modules, and executing in cooperation with a sandbox of one of the plurality of swappable pods or cards.	
4. The platform of claim 1, further comprising: at least one of the one or more swappable pods or cards having a plurality of types of memory and memory superposition with selectable arrangement of the plurality of types of memory as multilevel cache, interleaved memory, swapped memory over a specified address range, and memory allocated to specified address ranges according to parameterized task needs.	Cragon discloses or renders obvious “at least one of the one or more swappable pods or cards having a plurality of types of memory and memory superposition with selectable arrangement of the plurality of types of memory as multilevel cache, interleaved memory, swapped memory over a specified address range, and memory allocated to specified address ranges according to parameterized task needs,” as shown in at least the following exemplary citations: <i>See e.g.</i>, Cragon at 73: Caches can be classified by the interconnection topology; the selection and interconnection of the various cache components. A multilevel cache is one that has a first-level cache (either instruction, unified or split) and a second-level unified cache. <i>See e.g.</i>, Cragon at 91: The idea at work with a multilevel cache is that a small and fast first-level cache is used and that the second level cache can (1) reduce the transport time to the first-level cache and (2) reduce the bus loading. With the proper interplay of these ideas, the cache system performance will be improved. Reduction in transport time is advantageous because on-chip caches were initially small with a resulting high Pmiss. Because the weighted time to reference an AU not in the cache is the product of Pmiss and transport time, these two parameters are kept in balance with a second-level cache. (Another form of multilevel cache, disk cache, is discussed in Section 4.2.4.) Bus-loading reduction is important for multiprocessor systems where a number of processors must communicate over one bus to the main memory.

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

A simple two-level unified cache system is shown in Figure 2.27. This system has caches L1 and L2 between the processor and the main memory.

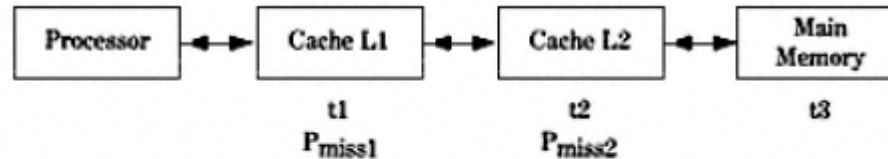


Figure 2.27
Two-level cache.

See e.g., Cragon at 110:

Given that real memory is large enough to hold a number of partitions, each partition is allocated its own address space in real memory. The monitor then manages these partitions, assigning space, adjusting addresses, and swapping partitions in and out. As jobs are completed, space in memory is vacated and the monitor attempts to load another partition. The problem with this system is primarily that memory becomes fragmented and is used inefficiently.

See e.g., Cragon at 191:

For a system that will have a number of tasks resident in memory at one time, there may be a number of bounds register pairs, one for each task. The register pairs are addressed by a pointer that is set when a task is switched in. The use of bounds registers assumes that a task, program, and/or data, is allocated within a contiguous address space. Bounds registers permit sharing between tasks of an address space by identifying overlapping spaces by the bounds registers.

See e.g., Cragon at 231:

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

Interleaved memories provide high-memory bandwidth by distributing the read or write accesses over a number of memory modules, which is a form of parallelism.

See e.g., Cragon at 232:

Interleaved memories are ideal for providing high bandwidth access to a long vector of data (either read or write). An interleaved memory requires two addresses: the address of the module and the address of the word within the module. Low-order interleaving generates these two addresses by dividing the memory address by the address interleaving degree n : the quotient is the word address presented to each module and the remainder selects the module. This division by an integer power of two is a trivial operation for designs having a degree of interleaving that is an integer power of two.

See e.g., Cragon at 232:

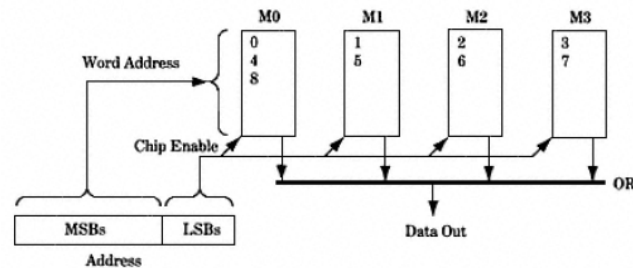


Figure 5.1
Four-way interleaved memory.

See e.g., Cragon at 257:

The efficiency problems of the batch monitor are addressed in software-managed memory systems by partitioned allocations, as shown in Figure 3.2(b). Given that real memory is large enough to hold a number of partitions, each partition is allocated its own address space in real memory. The monitor then manages these partitions, assigning space, adjusting addresses, and swapping partitions in and out. As jobs are completed, space in

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

memory is vacated and the monitor attempts to load another partition. The problem with this system is primarily that memory becomes fragmented and is used inefficiently.

See e.g., Cragon at 414-415.

For paged systems, protection is at the page or block level and for segmented systems at the segment level. For example, the segment length field of a segment table operates in the fashion of a bounds register. An example of this is the block translation system of the PowerPC 601 that has a block partition of memory that is similar to a segment. A block name is presented to the block TLB and is checked for being within the starting address and the length of the block. Block length is encoded with six bits, rather than having an upper address, in the BAT registers discussed previously. The length of the block is a binary progression: 128 Kbytes, 256 Kbytes, 512 Kbytes, . . . 8 Mbytes, an encoding designed to facilitate length checking. Memory keys specify the type of access that is permitted to a memory location, page, or segment. Typical access types are: Unlimited, Read Only, Execute Only, and/or a task I.D. The executing program key is usually found in the processor status register that, in combination with the type of access, is compared to the "lock" associated with the memory and found in the name translation system. If the key matches the lock, the requested access is performed. There is usually a skeleton key that can be used by the operating system to access all of memory regardless of the key associated with each of the keyed memory spaces.

See e.g., Cragon at 491:

Interleaved memories provide high-memory bandwidth by distributing the read or write accesses over a number of memory modules, which is a form of parallelism. The array disk to be described in Section 5.2.3 is another instance of memory interleaving. Early research into interleaving was concurrent with and a "back up" to hierarchical memory research. In the late 1950s and 1960s no one was sure if cache technology development would be successful.

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

[7.pre] A method for data flow processing in a platform, comprising:	<i>See claim element 1.pre.</i>
[7.a]. configuring a plurality of swappable pods or cards in one or more chassis, coupled through a messaging interface network, to implement a data flow processing architecture;	<i>See claim elements 1.pre and 1.a.</i>
[7.b] configuring user-definable hardware modules or user-definable software modules in each of one or more of the plurality of swappable pods or cards;	<i>See claim element 1.c.</i>
[7.c] communicating a data flow in accordance with the implemented data flow processing architecture among one or more hardware modules or one or more software modules in each of the plurality of swappable pods or cards, including the user-definable hardware	<i>See claim elements 1.c and 1.d.</i>

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

modules or user-definable software modules; and	
[7.d] selecting a memory superposition in at least one of the plurality of swappable pods or cards having a plurality of types of memory with selectable arrangement of the plurality of types of memory as multilevel cache, interleaved memory, swapped memory over a specified address range, and memory allocated to specified address ranges according to parameterized task needs.	<i>See claim 4.</i>
8. The method of claim 7, wherein the communicating the data flow comprises: communicating the data flow through the messaging interface network, comprising a network coupled to the	<i>See claim element 1.e.</i>

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

plurality of swappable pods or cards and supporting messaging-based communication using packets each having a header with a chassis identifier, a board identifier, a module identifier, an instance identifier, and a type identifier, so that each type of module, each instance of a type of module, and each module on each board in each chassis can be addressed through the header.	
9. The method of claim 7, wherein the communicating the data flow comprises: bypassing the messaging interface network and communicating through a direct connection among two or more pods, cards or components.	<i>See claim 2.</i>

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

10. The method of claim 7, further comprising: allocating at least one processor of a processor card, from among the plurality of swappable pods or cards, to offloading hardware tasks to software, orchestrating data flows, orchestrating configurations of hardware modules, or executing in cooperation with a sandbox of one of the plurality of swappable pods or cards.	<i>See claim 3.</i>
[13.pre] A tangible, non-transitory, computer-readable media having instructions thereupon which, when executed by a processor, cause the processor to perform a method comprising:	<i>See claim elements 1.pre and 1.a-1.e.</i>
[13.a] configuring a plurality of swappable pods or cards in one or more chassis, coupled through a messaging	<i>See claim elements 1.a-1.c.</i>

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

interface network, to implement a data flow processing architecture;	
[13.b] configuring user-definable hardware modules or user-definable software modules in each of one or more of the plurality of swappable pods or cards;	<i>See</i> claim element 1.c.
[13.c] communicating a data flow in accordance with the implemented data flow processing architecture among one or more hardware modules or one or more software modules in each of the plurality of swappable pods or cards, including the user-definable hardware modules or user-definable software modules; and	<i>See</i> claim elements 1.c and 1.d.
[13.d] communicating the data flow through the messaging interface network, comprising a network coupled to the	<i>See</i> claim element 1.e.

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

plurality of swappable pods or cards and supporting messaging-based communication using packets each having a header with a chassis identifier, a board identifier, a module identifier, an instance identifier, and a type identifier, so that each type of module, each instance of a type of module, and each module on each board in each chassis can be addressed through the header.	
14. The computer-readable media of claim 13, wherein the method further comprises: allocating at least one processor of a processor card having a plurality of processors, from among the plurality of swappable pods or cards, to offloading hardware tasks to software, orchestrating data flows, orchestrating	<i>See claim 3.</i>

Exhibit 753-08
Invalidity of the '753 Patent by Cragon

configurations of hardware modules, or executing in cooperation with a sandbox of one of the plurality of swappable pods or cards.	
15. The computer-readable media of claim 13, wherein the method further comprises: selecting a memory superposition in at least one of the plurality of swappable pods or cards having a plurality of types of memory with selectable arrangement of the plurality of types of memory as multilevel cache, interleaved memory, swapped memory over a specified address range, and memory allocated to specified address ranges according to parameterized task needs.	<i>See claim 4.</i>