Internal distribution code:
(A) [ ] Publication in OJ
(B) [ ] To Chairmen and Members
(C) [X] To Chairmen
(D) [ ] No distribution

Datasheet for the decision of 18 January 2011

Case Number:	T 0263/07 - 3.5.06
Application Number:	00119272.3
Publication Number:	1056016
IPC:	G06F 13/10
Language of the proceedings:	EN

Title of invention:
Application programming interface for data transfer and bus management over a bus structure

Applicant:
SONY ELECTRONICS, INC.

Headword:
Application programming interface/SONY

Relevant legal provisions:
RPBA Art. 4, 5, 15
EPC R. 100(2)

Relevant legal provisions (EPC 1973):
EPC Art. 84, 56

Keyword:
"Claims - clarity (no)"
"Inventive step - no (all requests)"
"Request for telephone interview with rapporteur - refused"

Decisions cited:
T 1109/02

Catchword:
Request for telephone interview with rapporteur refused - see Reasons 2.1. - 2.3.

Europäisches Patentamt
Beschwerdekammern

European Patent Office
Boards of Appeal

Office européen des brevets
Chambres de recours

Case Number: T 0263/07 - 3.5.06

DECISION
of the Technical Board of Appeal 3.5.06
of 18 January 2011

Appellant:

SONY ELECTRONICS, INC.
1 Sony Drive
Park Ridge
New Jersey 07656-8003 (US)

Representative:

Molyneaux, Martyn William
Langner Parry
1-2 Bolt Court
London EC4A 3DQ (GB)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted 20 September 2006 refusing European patent application No. 00119272.3 pursuant to Article 97(1) EPC 1973.

Composition of the Board:

Chairman: D. H. Rees
Members: A. Teale
M.-B. Tardo-Dino

Summary of Facts and Submissions

I. The appeal is against the decision by the examining division dispatched on 20 September 2006 to refuse European patent application 00119272.3 on the basis that the subject-matter of the claims according to both the main and the auxiliary request (both received on 17 October 2005) did not involve an inventive step, Article 56 EPC 1973, in view of the following document:

D1: "PC Intern 4 Systemprogrammierung", M. Tischer, pages 162 to 181, Data Becker GmbH, 1994, Düsseldorf, DE.

II. A notice of appeal was received on 1 November 2006, the appeal fee being paid on 6 November 2006.

III. With a statement of grounds of appeal, received on 19 January 2007, the appellant filed a copy of the following document:

"FreeBSD Developers' Handbook", section 9.1 "DMA: What it is and How it Works", 11 pages.

The appellant requested oral proceedings in the event that the board was minded to refuse the appeal.

IV. The board issued a summons to oral proceedings. In an annex to the summons the board set out its preliminary opinion on the appeal, in particular raising clarity objections against the claims according to the main and auxiliary requests on which the appealed decision was based and questioning the inventive step of the claimed subject-matter in view of the prior art acknowledged in the application itself (referred to as "Skipstone" below).

V. With a submission received on 17 December 2010 the appellant filed an amended page 6 of the description and new sets of claims according to a new main request and a new auxiliary request I. The appellant also requested that the previous main and auxiliary requests be re-labelled to become auxiliary requests II and III, respectively, and that the board amend the expression in claim 3 according to auxiliary requests II and III "an a node" to read "and a node". The appellant moreover requested that a patent be granted based on the claims in the main request and, if this could not be allowed, on the basis of the claims according to each of auxiliary requests I to Ill, in that order. The appellant also stated that it would not be represented at the oral proceedings and that the oral proceedings should take place in its absence, a decision being taken on the basis of the requests and arguments contained in the submission.

VI. On 10 January 2011 a letter was received from the appellant requesting that the rapporteur telephone the appellant's representative to discuss whether the application could be found in order for grant in accordance with one of the appellant's requests prior to the oral proceedings so that the oral proceedings could be cancelled.

VII. In a communication dated 11 January 2011 the board informed the appellant that the date for oral proceedings was maintained.

VIII. Claim 1 according to the main request reads as follows:

"A method of providing a memory-mapped interface to an application (24) having one or more buffers and managing high-speed asynchronous data transfer operations between the application (24) buffers and a serial bus structure (28) comprising the steps of:

a. receiving a request through an applications interface (20) for transfer of a block of data from the application (24) wherein the request includes an address for an application buffer, a starting address in an address space of the bus structure (28), a length of data to be transferred and a direction of the transfer;

b. in response to a command from the applications interface (20) to an automatic transaction generator (38), automatically generating the multiple read or write high-speed serial bus transactions necessary to complete the transfer of the block of data across the serial bus structure (28) without direct processor control of a processor corresponding to the application; and

c. notifying the application (24) when the data transfer is complete."

Claim 1 according to auxiliary request I differs from that according to the main request only in that in feature "b" the expression "command" has been replaced by "single communication" and that after the expression "automatic transaction generator (38)" the expression "independent of the application" has been inserted.

Claim 1 according to auxiliary request II only differs from that according to the main request in that the expression "of a processor corresponding to the application" has been deleted.

Claim 1 according to auxiliary request III only differs from that according to auxiliary request II in that the term "single" has been inserted before the expression "command".

Each request also comprises an independent claim 3 directed to "a memory-mapped interface".

IX. Oral proceedings were held on 18 January 2011 in the absence of the appellant, as announced in advance.

The board understood the appellant's substantive requests to be as follows: that the decision under appeal be set aside and that a patent be granted on the basis of the new main request or alternatively the auxiliary request I both filed with the letter of 17 December 2010, or alternatively on the basis of the auxiliary request II or auxiliary request Ill, respectively, former main request and auxiliary request filed before the examining division (12 October 2005), with the linguistic corrections requested in the letter of 17 December 2010 and page 6 of the description submitted with the letter of 17 December 2010, with the description and drawings otherwise as originally filed.

X. At the end of the oral proceedings the board announced its decision.

Reasons for the Decision

1. *The admissibility of the appeal*

In view of the facts set out at points I to III above, the appeal is admissible, since it complies with the EPC formal admissibility requirements.

2. *The appellant's request that the rapporteur telephone the representative to discuss the chances of the pending requests being allowed*

2.1 In the letter received on 10 January 2011 the appellant essentially requested a telephone interview with the rapporteur to discuss the allowability of the requests on file. As established in the case law of the boards of appeal, as a matter of principle, the EPC foresees the absolute right to oral proceedings under Article 116(1) EPC 1973, but not the right to a telephone interview (cf. Case Law of the Boards of Appeal of the EPO, 6th edition, 2010, VII.B.2.7.2 concerning the department of first instance, in particular).

2.2 As to appeal proceedings more specifically, Articles 4 and 5 RPBA (Rules of Procedure of the Boards of Appeal of the European Patent Office, OJ EPO 2007, 536, the wording of which remains unchanged after the entry into force of EPC 2000) provide that certain steps in the proceedings may be taken by the rapporteur. Where this is the case the rapporteur's duties consist of either ensuring, under the board's supervision, that the procedural rules or the directions of the board of appeal are complied with by the parties, or, where it comes to substantive matters (Article 5(3) RPBA), of acting on behalf of the board. This, in other words, implies that the other members of the board have been informed and put in the position to give an informed opinion on the action to be taken. To this end it is important that the same case is presented to all of the board's members. If one of the board's members were privy to evidence or arguments not available to the other members then this would be a breach of the principle of collective decision making and would be in conflict with Article 21 EPC 1973; see T 1109/02 (not published in OJ EPO, reasons, point 1).

2.3 Since the requested telephone interview could have led the rapporteur to take a position on an issue where a collective decision would have been required, or to commit the board without preliminary discussion, the request was refused as not being compatible with the above mentioned principle and rules governing appeal proceedings.

2.4 A further communication by the board after the summons to oral proceedings was not necessary and had also not been requested by the appellant. Under Rule 100(2) EPC (corresponding to Article 110(2) EPC 1973 in conjunction with Rule 66 (1) EPC 1973) the board shall invite the parties "as often as necessary" to file observations. In the present case oral proceedings were arranged as requested by the appellant and because it was the most efficient procedural course of action to be taken at this stage. The purpose of oral proceedings is to give the party the opportunity to present its case and to be heard. However a party gives up that opportunity if it does not attend the oral proceedings. By filing amended claims before the oral proceedings and then not attending those oral proceedings the appellant must also expect a decision based on objections which may be raised against such claims in its absence, Article 15(3,6) RPBA. In the present case the board had already raised objections regarding *inter alia* clarity and inventive step against the claims then on file (now auxiliary requests II and III) in the annex to the summons to oral proceedings. Since essentially the same objections also applied to the claims of the appellant's new main request and auxiliary request I, the board considered a further communication to be unnecessary.

3. *The appellant's non-attendance at the oral proceedings*

3.1 As announced in advance, the duly summoned appellant did not attend the oral proceedings and requested that a decision be taken at the oral proceedings on the basis of the requests and arguments contained in the submission received on 17 December 2010.

3.2 In accordance with Article 15(3) RPBA, the board relied for its decision only on the appellant's written submissions. The board was in a position to decide at the conclusion of the oral proceedings, since the case was ready for decision (Article 15(5, 6) RPBA), and the voluntary absence of the appellant was not a reason for delaying a decision (Article 15(3) RPBA). Moreover the appellant had explicitly requested that a decision be taken in its absence at the oral proceedings.

4. *The document "The FreeBSD Developers' Handbook"*

4.1 In the annex to the summons to oral proceedings the board stated that, as DMA (direct memory access) had been discussed in the first instance proceedings, it seemed that this document could have been presented before the examining division. The board also questioned the date on which this document was written and whether its content reflected common general knowledge at the priority date. Since this document did not appear to relate to the case under appeal, Article 12(4) RPBA, the board was not inclined to admit it into the proceedings.

4.2 The appellant has responded that, because it had already been asserted that DMA was well known in the art at the priority date, the handbook was being used to provide a more detailed explanation of DMA.

4.3 The board is not convinced by the appellant's arguments. Firstly, the general perception of the DMA approach may have developed over time so that it is necessary to establish the date of any description of DMA. The document is however unclear on this point, since it states explicitly that it was last updated on 8 October 1997, more than 20 months after the priority date. Secondly, the appellant has not provided any explanations as to why the document could not have been presented during the discussion of DMA before the examining division. Consequently the appellant has not persuaded the board to deviate from its provisional opinion that this document does not relate to the case under appeal.

4.4 Hence the board does not admit this document into the proceedings, Article 12(4) RPBA.

5. *The context of the invention*

5.1 The invention relates to an application programming interface (API) for applications to communicate over a serial bus structure in an asynchronous data format, for instance that defined in the IEEE 1394 standard; see page 1, line 13, to page 2, line 8, of the description. Such bus structures may be used to carry digital video signals between devices such as video cameras, VCRs and computers. The API comprises a collection of software routines which are called by an application to manage data being written to and obtained from a device over the bus. As shown in figure 2, in each device connected to the serial bus, applications communicate with the API which in turn communicates with the hardware and physical interface connected to the serial bus structure.

5.2 The API provides a memory-mapped interface to each application for asynchronous data transfers, in the case of the IEEE 1394 standard the bus structure providing a 64 bit address space; see page 15, lines 6 to 8, of the description. Data transfers over the serial bus are completed by "transactions". Read transactions involve data in a buffer associated with an application being written to a certain area of the IEEE 1394 bus address space. Correspondingly, write transactions involve data coming from a certain area of the IEEE 1394 bus address space being written to a buffer associated with an application; see page 15, line 11, to page 16, line 2, of the description.

5.3 According to the invention, an automatic transaction generator is used to automatically generate the transactions necessary to complete the data transfer without direct control by the processor of the application or supervision by the API, thus reducing the loading on the application processor; see page 4, lines 22 to 26, of the description. The API defines essentially a direct memory access (DMA) model (see section 6 below), utilizing a level of hardware automation to automatically generate the requests necessary to complete the transfer, the automatic transaction generator (38) forming part of the hardware and physical interface (26); see page 8, lines 7 to 10, and page 14, lines 2 to 6, of the description and figure 2. However the automatic transaction generator can also be implemented in software within the API; see page 14, lines 19 to 22 (see section 7 below regarding clarity).

6. *The common general knowledge at the priority date*

6.1 The appellant has stated that "the DMA principle is an old principle" and has not disputed that the DMA technique, as exemplified by D1, was common general knowledge at the priority date (2 February 1996) of the present application.

6.2 D1 relates to the DMA controller typically found on a PC mother board and its use, for instance, to transfer data between a floppy disk and memory; see pages 162 to 163, in particular the section "Zusammenspiel zwischen Hardware und Software bei DMA-Transfers". When the appropriate BIOS function is called by an application the parameters of the DMA data transfer, such as the number of bytes to transfer, are stored in registers of the DMA controller; see the paragraph bridging pages 166 and 167, in particular the "counter register". Once these parameters have been set up, the transfer can be triggered by activating one of the "DMA Request" lines of the DMA controller; see "DREQ0-DREQ3" in the table on page 165 and page 171, lines 23 to 29. Once a DMA request has been made, the DMA controller sends a "Hold Request" (see "HRQ" in the table on page 165) to the CPU which responds with the "Hold Acknowledge" signal (see "HLDA" in the table on page 165) and isolates itself from the system buses; see page 171, lines 30 to 31. The DMA controller then controls the buses and automatically generates the necessary bus instructions to transfer the desired data.

6.3 The appellant has argued that D1 does not disclose an automatic transaction generator. The board disagrees. The bus instructions generated by the DMA controller in D1 to carry out a data transfer can be regarded as transactions. Moreover the DMA controller known from D1 can be seen as automatic, since, once activated by a DMA request, it transfers the sequence of bytes defined by the parameters of the DMA data transfer automatically without the intervention of the processor.

6.4 The appellant has also argued that the DMA controller known from D1 is not separate/independent from the application itself, the DMA controller only functioning to manage the transfer of received transactions by independently controlling the bus. The board does not accept this argument, since, as set out at point 6.2 above, an application does not communicate directly with the DMA controller. Instead an application calls BIOS functions to access the DMA controller; see the sentence bridging pages 162 and 163. Moreover, as stated above, once activated by a DMA request, the DMA controller generates the instructions to transfer the sequence of bytes defined by the parameters of the DMA data transfer automatically without the intervention of the processor.

7. *Clarity, Article 84 EPC 1973*

7.1 *The question of which processor claim 1 refers to*

7.1.1 Claim 1 according to what are now auxiliary requests II and III refers to a processor. In the annex to the summons to oral proceedings the board raised a clarity objection *inter alia* against this claim that it was not clear which processor was being referred to, since in the video system shown in figure 3 each of a video camera, a VCR and a computer contains a CPU; see figure 4 and page 6, lines 11 to 15.

7.1.2 The appellant has not amended these claims or submitted any counter-arguments. The board therefore sees no reason to deviate from its preliminary opinion that, in the context of the description and drawings, it is unclear in claim 1 according to auxiliary requests II and III which processor is being referred to.

7.2 *The expression in claim 1 "without direct processor control"*

7.2.1 Claim 1 according to the appellant's main request and auxiliary requests I to III sets out the automatic transaction generator, in response to a command from the applications interface, automatically generating serial bus transactions "without direct processor control", claim 1 according to auxiliary request I setting out the additional qualification that the automatic transaction generator is independent of the application.

7.2.2 In the annex to the summons to oral proceedings the board raised a clarity objection against *inter alia* claim 1 according to what are now auxiliary requests II and III that, according to the description (see page 14, lines 19 to 22), the automatic transaction generator could also be implemented in software within the API. In such a case it was unclear in claim 1 how the automatic transaction generator could automatically generate serial bus transactions "without direct processor control", since it appeared that a software implementation would rely on the processor.

7.2.3 The appellant has responded that the application does not state that the automatic transaction generator does not rely on the processor. It merely states that the automatic transaction generator generates the needed transactions "without direct processor control". As a result, the automatic transaction generator is able to access the processor or a separate processing element dedicated to the automatic transaction generator. For example, in one embodiment in which the automatic transaction generator is implemented as software, it is able to utilize the processor to generate the needed transactions by controlling/monitoring the bus thereby freeing the API and application to perform other functions. Indeed, this is because relying on the processor does not require direct processor control, rather multiple elements can access the processor simultaneously despite none having direct control over the processor. Thus it is clear that the automatic transaction generator is able to rely on the processor without direct control over the processor (letter of 17 December 2010, Annex page 2).

7.2.4 The board is not convinced by the appellant's arguments. Starting with the appellant's last statement, the expression "direct processor control" is used in the application to mean control **by** the processor, not control **over** the processor, as is implied by the appellant's last statement. The description refers to the automatic transaction generator being used to automatically generate the transactions necessary to complete the data transfer without direct processor control or supervision by the applications programming interface; see page 4, lines 22 to 26, and page 6, line 29, to page 7, line 2, of the description. Moreover the application does not disclose the automatic transaction generator automatically generating serial bus transactions without direct processor control whilst still relying on the processor in the sense of accessing it. In the board's view an application or the API running on the processor is inevitably under direct processor control. Furthermore the application does not disclose a separate processing element dedicated to the automatic transaction generator. Also, even if it were disclosed in the application, utilizing the processor to generate the needed transactions by controlling/monitoring the bus thereby freeing the API and application to perform other functions, as the appellant has argued, would not be without direct processor control.

7.2.5 The board concludes that claim 1 according to the appellant's main request and auxiliary requests I to III is unclear, contrary to Article 84 EPC 1973.

7.2.6 For the above reasons none of the appellant's main request and auxiliary requests I to III is allowable. The appeal must therefore be dismissed. However the board further notes that, assuming the skilled person were to understand the claims as set out below, the claimed subject-matter also does not involve an inventive step, Article 56 EPC 1973, as follows.

8. *The prior art acknowledged in the application ("Skipstone")*

8.1 The description (see page 3, line 1, to page 4, line 8) describes an application programming interface (API) for applications using the IEEE 1394 standard serial bus developed by Skipstone, Inc. The appellant has argued that this disclosure constitutes the closest prior art available (statement of grounds, page 2). The board accepts that Skipstone forms an appropriate starting point for assessing inventive step. According to page 3, line 18, to page 4, line 2, "During asynchronous data transfers, the Skipstone API actively manages the required transactions to complete the data transfer. During an asynchronous incoming write transaction, the application provides a buffer to the API, mapped to a certain area of the 1394 bus address space. As write transactions arrive at the API, their data is written to the buffer. During an asynchronous incoming read transaction the application is responsible for making sure that the buffer contains useful information. The 1394 bus driver then reads the data from the buffer at the requested address when the read transaction arrives. For both write and read transactions, the Skipstone API actively manages and generates each necessary transaction. For example, if a block of data is being transferred to the application, of a size requiring multiple transactions, the Skipstone API requires the application to describe each 1394 transaction necessary to complete the transfer of the block of data. This consumes significant overhead by the processor of the application as well as the full attention of the API during an asynchronous data transfer operation."

8.2 It is implicit in Skipstone that the write and read transactions specify the address of the application buffer and that a bus interface circuit is present in order for the interface to work.

8.3 It is common ground between the board and the appellant that Skipstone does not disclose an automatic transaction generator. Instead, as stated at point 8.1 above, the Skipstone API requires the application to describe each transaction necessary to complete the transfer of the block of data; see page 3, lines 26 to 29.

9. *Novelty, Article 54(1,2) EPC 1973*

9.1 *Main request*

9.1.1 In terms of claim 1 according to the main request, Skipstone discloses a method of providing a memory-mapped interface to an application having one or more buffers and managing high-speed asynchronous data transfer operations between the application buffers and a serial bus structure comprising the steps of: receiving a request through an applications interface for transfer of a block of data from the application wherein the request includes an address for an application buffer and a direction of the transfer and generating the multiple read or write high-speed serial bus transactions necessary to complete the transfer of the block of data across the serial bus structure.

9.1.2 The subject-matter of claim 1 differs from Skipstone in that:

i. the request received through the applications interface also includes a starting address in an address space of the bus structure and a length of data to be transferred;

ii. an automatic transaction generator functions in response to a command from the applications interface, automatically and without direct processor control of a processor corresponding to the application and

iii. the application is notified when the data transfer is complete.

9.2 *Auxiliary request I*

9.2.1 Claim 1 differs from that according to the main request in that the expression "command" has been replaced by "single communication" and that after the expression "automatic transaction generator (38)" the qualification "independent of the application" has been inserted.

9.2.2 Hence the subject-matter of claim 1 differs from Skipstone in difference features "i" and "iii" set out above for the main request and in that:

ii.1. an automatic transaction generator independent of the application functions in response to a single communication from the applications interface, automatically and without direct processor control of a processor corresponding to the application.

9.3 *Auxiliary request II*

9.3.1 Claim 1 differs from that according to the main request in that the expression "of a processor corresponding to the application" has been deleted.

9.3.2 Hence the subject-matter of claim 1 differs from Skipstone in difference features "i" and "iii" set out above for the main request and in that:

ii.2. an automatic transaction generator functions in response to a command from the applications interface, automatically and without direct processor control.

9.4 *Auxiliary request III*

9.4.1 Claim 1 only differs from that according to auxiliary request II in that the term "single" has been inserted before the expression "command".

9.4.2 Hence the subject-matter of claim 1 differs from Skipstone in difference features "i" and "iii" set out above for the main request and in that:

ii.3. an automatic transaction generator functions in response to a single command from the applications interface, automatically and without direct processor control.

9.5 *Conclusion on novelty*

It follows from the above analysis that the subject-matter of claim 1 according to the appellant's main request and auxiliary requests I to III is new, Article 54(1,2) EPC 1973, having regard to Skipstone.

10. *Inventive step, Article 56 EPC 1973*

10.1 *Approach to assessing inventive step*

10.1.1 The board finds that difference features "i", "ii"/"ii.1"/"ii.2"/"ii.3" and "iii", set out above for the appellant's main request and auxiliary requests I to III, have no surprising combined synergistic effect, so that their individual contributions to inventive step must be considered separately. Feature "i" in each case concerns the composition of the request received through an application interface, whilst features "ii"/"ii.1"/"ii.2"/"ii.3" all concern the subsequent automatic generation of serial bus transactions. Finally, feature "iii" in each case concerns what happens once the data transfer is complete. Consequently the difference features in each case are technically unrelated and all produce the effect that they would produce alone, there being no surprising combined synergistic effect. The appellant has not argued that there is any such effect.

10.1.2 As explained in more detail below for each of the appellant's main request and auxiliary requests I to III, the skilled person starting from Skipstone and, as a matter of usual design, seeking to speed up operation, would have recognised that transaction generation by the interaction of the application and the API placed an undue burden on the processor and would have applied the DMA principle, a matter of common general knowledge, to solve this problem, thus adding features "i", "ii"/"ii.1"/"ii.2"/"ii.3", as the case may be, and "iii" to arrive at the claimed subject-matter without inventive step.

10.1.3 While avoiding engaging in an *ex post facto* analysis, the board notes that the description (see page 14, lines 2 to 6) confirms that the alleged invention relates to the application of the DMA principle to the Skipstone disclosure.

10.2 *The main request*

10.2.1 Difference feature "i" (the request received through the applications interface also including a starting address in an address space of the bus structure and a length of data to be transferred) concerns usual matters of design which were common general knowledge (see, for instance, D1, page 163, lines 3 to 14) at the priority date and which the skilled person would have introduced to fill in the gaps in the Skipstone disclosure without inventive step.

10.2.2 Regarding difference feature "i", the appellant has argued that because in Skipstone the application handles the transaction generation and the API merely manages the transfers, the skilled person would have had no motivation to include the starting address and length. Such an addition would have been superfluous and would thus have involved an inventive step. The board is not convinced by these arguments. The skilled person starting from Skipstone and seeking to speed up operation by applying the DMA principle to reduce processor loading would have delegated functionality from the application and API to the automatic transaction generator. Part of this delegation would have necessarily involved providing the automatic transaction generator with the basic information necessary for generating the required transactions, namely a starting address in an address space of the bus structure and a length of data to be transferred.

10.2.3 As to difference feature "ii" (an automatic transaction generator functioning in response to a command from the applications interface, automatically and without direct processor control of a processor corresponding to the application), in applying the DMA principle to Skipstone the skilled person would have provided dedicated hardware for automatic bus transaction generation without direct control by the application processor to reduce the loading on the application processor. The operation of the transaction generator in response to a command from the API, rather than having to constantly involve the API and application, would have been a necessary measure to reduce the loading on the processor to allow it to carry out other tasks during a serial bus data transfer and would have been the usual choice, given that in Skipstone the API acts as an interface between the application and the hardware and physical interface. Thus feature "ii" does not contribute to inventive step either.

10.2.4 Regarding difference feature "ii", the appellant has argued that the DMA principal has been misconstrued. DMA controllers were designed to transfer data between a peripheral device and the memory bypassing the CPU; see D1, page 162, first paragraph. Hence DMA principles did not relate to transaction generation, but instead were solely directed to transfer management. Thus applying the DMA principle to speed up operation would only have resulted in using a DMA controller to offload the transfer duties from the API, not in using the claimed automatic transaction generator. The skilled person would simply have left the transaction generation to the application itself as taught by Skipstone. The board does not accept these arguments. As stated at point 6.3 above, the bus instructions generated by the DMA controller to carry out a data transfer can be regarded as transactions. Moreover the skilled person starting from Skipstone would have recognised that not only the transfer duties of the API loaded the processor, but also the generation of transactions by the application. Both activities would have been delegated to an automatic transaction generator, in application of the DMA principle, to reduce the processor loading without inventive step.

10.2.5 Turning to difference feature "iii", given that the purpose of applying the DMA principle to Skipstone would have been to allow the processor to carry out other tasks during a serial bus data transfer, the skilled person would have necessarily modified Skipstone to notify the processor upon completion of the transfer of the block of data to avoid the need for the processor to monitor the status of the data transfer. Moreover both the application and the API run on the processor, and the notification of the application by the API would be the usual route for informing an application of an interface-related matter. Hence difference feature "iii" does not contribute to inventive step either.

10.2.6 Regarding difference feature "iii", the appellant has argued that, since in Skipstone the application would have known that the transfer was complete due to it being responsible for the generation of the needed transactions, there would have been no motivation to modify Skipstone to notify the application when the data transfer was complete even in the light of D1 so that this difference feature would have involved an inventive step. The board does not accept this argument. Once transaction generation had been delegated from the application and API, the skilled person would have recognized that the application would no longer know that the transfer was complete, thus necessitating notifying the application once this had occurred.

10.3 *Auxiliary request I*

10.3.1 For the reasons set out for the main request, difference features "i" and "iii" do not contribute to inventive step.

10.3.2 Regarding the expression in claim 1 "single communication", the appellant has argued that, unlike "commands" which can include multiple sub-commands and still be considered a single command depending on perspective, a single communication represents a single data transfer regardless of the number of sub-commands included in that single communication. D1 did not teach a "single communication" because the triggering of a block transfer by a signal on one of the DREQx-lines did not trigger the generation of multiple transactions. The board is not convinced by this argument that the expression "single communication" has any different technical meaning to "command" or "single command", since these general expressions are not used with any specific meanings in the description and drawings. Also the board does not agree that in D1 triggering of a block transfer by a signal on one of the DREQx-lines does not trigger the generation of multiple transactions. According to the paragraph in D1 bridging pages 166 and 167, a DMA transfer involves decrementing a counter register every time a byte is transferred, such a transfer constituting a "transaction" in the meaning of the claims.

10.3.3 Regarding difference feature "ii.1", in spite of the use of the expression "single communication" instead of "command" and the qualification that the automatic transaction generator is "independent of the application" the board finds that the reasoning given above in relation to feature "ii" (main request) also applies to feature "ii.1" *mutatis mutandis*. The skilled person applying the DMA principle to Skipstone would have triggered the automatic transaction generator with a signal (such as the DREQx signal in D1) falling under the term "single communication" as a matter of usual design; also see point 10.3.2 above. Furthermore transaction generation by the automatic transaction generator independently of the application would have been the aim of the skilled person seeking to apply the DMA principle to Skipstone to reduce the loading of the processor. Hence difference feature "ii.1" also does not contribute to inventive step.

10.4 *Auxiliary request II*

10.4.1 For the reasons set out for the main request, difference features "i" and "iii" do not contribute to inventive step.

10.4.2 Since difference feature "ii.2" differs from difference feature "ii" (main request) only in that it has been broadened by deleting the expression relating to the processor "corresponding to the application", the board finds that the reasoning given above in relation to feature "ii" also applies to feature "ii.2". Hence difference feature "ii.2" also does not contribute to inventive step.

10.5 *Auxiliary request III*

10.5.1 For the reasons set out for the main request, difference features "i" and "iii" do not contribute to inventive step.

10.5.2 Regarding difference feature "ii.3", compared to difference feature "ii.2" (auxiliary request II), in spite of the use of the expression "single command" instead of "command" the board finds that the reasoning given above in relation to feature "ii.2" also applies to feature "ii.3" *mutatis mutandis*. The skilled person applying the DMA principle to Skipstone would have triggered the automatic transaction generator with a signal (such as the DREQx signal in D1) falling under the term "single command" as a matter of usual design; also see point 10.3.2 above. Hence difference feature "ii.3" also does not contribute to inventive step.

11. *Conclusion*

Claim 1 according to the appellant's main request and auxiliary requests I, II and III is unclear, Article 84 EPC 1973. Moreover the subject-matter of the same claims does not involve an inventive step, Article 56 EPC 1973. For either of these reasons none of the requests is allowable.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:	The Chairman:
B. Atienza Vivancos	D. H. Rees