

Internal distribution code:

- (A) Publication in OJ
(B) To Chairmen and Members
(C) To Chairmen
(D) No distribution

**Datasheet for the decision
of 25 January 2008**

Case Number: T 0827/05 - 3.5.01

Application Number: 98116477.5

Publication Number: 0905618

IPC: G06F 9/46

Language of the proceedings: EN

Title of invention:

Microcontroller, data processing system and task switching
control method

Applicant:

MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.

Opponent:

-

Headword:

Task switching/MATSUSHITA

Relevant legal provisions:

-

Relevant legal provisions (EPC 1973):

EPC Art. 56

Keyword:

"Inventive step - (yes) after amendment"

Decisions cited:

-

Catchword:

-



Case Number: T 0827/05 - 3.5.01

D E C I S I O N
of the Technical Board of Appeal 3.5.01
of 25 January 2008

Appellant: MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.
1006, Oaza Kadoma
Kadoma-shi,
Osaka 571-8501 (JP)

Representative: Grünecker, Kinkeldey
Stockmair & Schwanhäusser
Anwaltssozietät
Leopoldstraße 4
80802 München (DE)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted 2 February 2005
refusing European application No. 98116477.5
pursuant to Article 97(1) EPC 1973.

Composition of the Board:

Chairman: S. Steinbrener
Members: W. Chandler
G. Weiss

Summary of Facts and Submissions

- I. This appeal is against the decision of the examining division to refuse the application on the ground that claims 1 to 16 did not involve an inventive step. The following documents were mentioned in the decision:
- D1: US-A-4 914 570
D2: US-A-5 655 146
- II. The appellant requested that the decision under appeal be set aside and that a patent be granted on the documents on file. The appellant also made an auxiliary request for oral proceedings. In the statement setting out the grounds of appeal, the appellant alleged various differences over the prior art and argued why D1 and D2 were not considered to suggest the invention.
- III. In the communication accompanying the summons to oral proceedings, the Board summarised the issues to be discussed and tended to agree with the examining division that the subject-matter of claim 1 lacked an inventive step over D2 and the skilled person's common general knowledge. In particular, the Board expressed the view that the alleged further differences did not follow from the wording of claim 1 on file.
- IV. In a response, the appellant filed new claims 1 to 13 and a post-published IEEE Article entitled "A 100mm² 0.95 W Single-Chip MPEG2 MP@ML Video Encoder with a 128GOPS Motion Estimator and a Multi-Tasking RISC-Type Controller", 45th ISSCC 1998, Digest of Technical Papers, pages 30 to 31 and 409 (hereinafter D3) to explain the invention. In a subsequent telephone call and letters,

the appellant made a reasoned request for postponement of the oral proceedings.

V. At the oral proceedings, which had been postponed according to the appellant's request, the appellant requested that the decision under appeal be set aside and that a patent be granted on the basis of claims 1 to 10 submitted at the oral proceedings. At the end of the oral proceedings the Chairman announced the decision.

VI. Claim 1 of the main request reads as follows:

"A data processing system comprising:

a plurality of hardware engines (111-115) forming a macro block pipeline, each of the hardware engines being assigned one task in a plurality of repetitive operations of encoding or decoding a data stream consisting of consecutive blocks of data, and

a microcontroller (101) for controlling said plurality of hardware engines, said microcontroller adopting an event-driven method capable of performing task switching in fast response to the occurrence of an event, said microcontroller including:

a processor (300) having a program counter (301) for sequentially executing said plurality of tasks;

a task management table (310) for storing task management information, including:

- (i) state information representative of the execution status of each task,
- (ii) priority information representative of the execution priority of each task,
- (iii) allocation information representative of the allocation of said plurality of tasks to said hardware engines, and
- (iv) program counter values for each task;

a plurality of register files (211-216) which can be used by said hardware engines as mutually independent working areas, and

a hardware scheduler (330) for allowing said processor to switch between said tasks without an interrupt handling routine, on the basis of said task management information,

said hardware scheduler (330) including:

- a) a determination unit (332) for identifying a task allocated to an execution-terminated hardware engine (111-115) on the basis of said task management information, when the execution of any one of said plurality of said hardware engines (111-115) is terminated;
- b) a state controller (331) which performs a function of updating of state information, upon being activated by said determination unit (332), and
- c) a selector (334) that reads out a program counter value for the task to be run next in the

processor (300), said task being determined by said state information and priority information and said value being set in the processor to start the execution of that task."

VII. The appellant argued essentially as follows:

The invention as claimed enabled efficient control of hardware engines in a pipeline environment using a hardware scheduler switching from one task to the next in an event-driven method without activating an interrupt handling routine in the operating system.

D2 concerned a central processor 100 operating in collaboration with auxiliary processors (coexecutors or COEXs) as shown in Figure 1. The interaction between the central processor and the COEXs was controlled by a host operation system 102 (cf. column 4, lines 30 to 41) that was responsible for the scheduling of the COEX operations. The host OS received requests for COEX operations, queuing the requests if necessary, and monitoring the state of the COEX (column 5, lines 5 to 20). When a COEX terminated its operation, this was reported by an I/O interrupt 106 to the central processor. Upon receiving an interrupt, the central processor provided this information to the host OS and the host OS carried out additional tasks for handling this event (cf. column 8, lines 40 to 62).

In order to execute an operation on a COEX, special program code was provided to the COEX through the main processor. A common data bus and a shared memory 107 was used for data transfer between the main processor and the COEXs. As only a single data transfer operation

between the main processor CEC and a COEX could be carried out at the same time, further data transfer requests of other COEXs had to be suspended until a first data access had finished. Additionally, a "COEX control program" was needed to manage the data access to the shared central electronic storage (cf. column 8, line 65 to column 9, line 2).

The information about the status of a COEX and the tasks scheduled for execution on a COEX were stored in the main memory of the system. Thus, the tasks of the host operation system for reading the task information from the main memory and determining an operation to be executed next on the COEX required additional computational resources.

Starting from prior art D2, a skilled person would arrive at neither an arrangement of the hardware engines in a pipeline structure nor a task switching on hardware basis in order to avoid additional operation system tasks. Hence, the present invention involved inventive step over this prior art.

The three-page article D3, published by the inventors in 1998 (after the priority date), discussed the claimed invention for a single-chip MPEG 2 video encoder. The interesting parts were page 30, right-hand column, first paragraph and page 31, table 1 and Fig. 3, being compatible with page 2, line 13 to page 3, line 4 of the application documents as filed. The implementation was discussed under the use of 4 states of the hardware engines. The important part was the hardware scheduler, which controlled the consecutive tasks by events, namely a state transition of tasks in

the hardware engines without using an interrupt handling routine.

Reasons for the Decision

1. As explained by the appellant (see point VII, above), the invention essentially concerns a data processing system (Figure 1) having a plurality of hardware engines (111 to 115) forming a macro block pipeline (e.g. for MPEG coding) controlled by a microcontroller (101) having a hardware scheduler (Figure 3: 330) switching from one task to the next without using an interrupt handling routine.
2. The examining division considered D2 to be the closest prior art. As also explained by the appellant, D2 discloses (Figure 1) a central processor operating in collaboration with auxiliary processors (coexecutors or COEXs). The interaction between the central processor and the COEXs is controlled by a host operating system (102) that schedules the COEX operations. When a COEX has terminated its operation, this is reported by an I/O interrupt (106) to the central processor. Upon receiving an interrupt, the central processor provides this information to the host OS, which carries out additional tasks for handling this event.
3. Refused claim 1 was very generally worded. However, it was common ground that it differed from D2 at least by the features that:

- a) the task management information included priority information representative of the execution priority of each task and was stored in a task management table;
 - b) the processor, task management table and the scheduler were in a microcontroller; and
 - c) the hardware engines were mutually connected to form a processing pipeline.
4. The examining division considered that these differences were well known or routine features in this field. The Board agrees that each of the concepts of task management, a hardware scheduler and pipeline processing are known *per se*.
5. However, present claim 1 is now more specific. Firstly, it clarifies the components of the system and their functions, namely that a data processing system contains the plurality of hardware engines forming the pipeline, each assigned a task in an encoding or decoding operation. Further, that the microcontroller for controlling the tasks has the task management table and a processor with a program counter. The system further comprises a plurality of register files that can be used by the hardware engines as mutually independent working areas. Finally, the hardware scheduler includes a determination unit for identifying a task when the execution of a hardware engine is terminated and activating a state controller that updates the state information for the task and a selector that reads out the program counter values of the next task to be run and sets it in the processor.

6. The Board considers that the distinguishing features solve the general problem of increasing the speed of parallel processing applications.

7. Although the features of the solution may essentially be individually known, the Board does not consider that the claimed solution is the inevitable result of the skilled person's common general knowledge about multitasking operating systems and pipeline processing. In particular, the Board considers that the skilled person would generally understand pipeline processing to be an arrangement in which the processes have a fixed timing relationship with each other, as in the well-known assembly line analogy. However, D2 concerns a conventional computing system in which a host program offloads work to the auxiliary processors (COEXs). The COEXs are essentially treated as flexible resources that execute code for variable length tasks provided by the main processor (via an address). Pending tasks requiring a COEX are queued until one is available as reported to the main processor by an I/O interrupt. In the Board's view, such a flexible system is at odds with the fixed timing generally inherent in pipeline processing so that the skilled person would, in the absence of further indications, not be inclined to incorporate pipeline processors in the variable length task structure of D2. The further changes to use the hardware registers and the specific features of the hardware scheduler allowing the processor to switch between tasks without an interrupt handling routine would require the system of D2 to be completely redesigned. The Board does not consider that the skilled person would envisage modifying and specifically optimising D2 to this extent.

8. Accordingly the Board judges that starting from D2, claim 1 can no longer be considered to lack an inventive step.

9. Apart from explicitly mentioning priority information and a task management table, document D1 is otherwise further from the invention because there is even more latency in interrupting and transferring tasks between processors than in D2. Thus for the same reasons as for D2, the Board does not consider that claim 1 lacks an inventive step starting from document D1.

10. The Board has also considered a further relevant document. This document is mentioned in the ISSCC paper (D3) that the appellant filed in response to the summons to oral proceedings. D3, written by, among others, the present inventors describes a single-chip MPEG2 video encoder called VDSP3 that corresponds to the present invention. The article states in the first line that the encoder VDSP3 has ten cores forming a macroblock-level pipeline "similar to that of a previous LSI, VDSP2 [2]". At the bottom of the page, reference [2] is given as "Toyokura, M., et al., 'Video DSP with a Macroblock-Level-Pipeline and a SIMD Type Vector-Pipeline Architecture for MPEG2 CODEC,' IEEE Journal of Solid-State Circuits, vol. 29, no. 12, Dec., 1994". The IEEE document is therefore prior art for the present application.

11. It is apparent, from this referenced document that the VDSP2 encoder has hardware engines in a macroblock pipeline. However, it appears from the description of the operation of these engines at page 1475, right hand

column, lines 7 to 10 as well as Figure 4(b):
"Permitted time for MB processing", and the performance estimation on page 1479, that the "processes", corresponding to some of the tasks in the invention, are assigned a fixed time, i.e. said permitted time for processing each macroblock, and thus have a fixed timing relationship with each other as mentioned above in connection with pipeline processing. Moreover, although page 1478, end of right-hand column, mentions "multitask control", it appears that the programmable control unit (DPCU) can only "judge the end of these processes" and is therefore not "event-driven" and does not respond directly to the end of a task as claimed. There is no mention of a hardware scheduler or its features.

12. It appears that the distinguishing features solve the problem of providing high-speed flexible task switching. However, in the Board's view the skilled person would not consider adding a flexible multitasking arrangement, especially the slower software solution from D2, and a hardware scheduler to the VDSP2 engine having fixed timing essentially for the same reasons given above why the skilled person would not consider a pipeline processing when starting from D2. In particular, the skilled person would not see the need for it when the execution time of the tasks is already roughly balanced and within the permitted time for a macroblock as shown in Figure 11 of the IEEE document.
13. Accordingly, the Board judges that the subject-matter of claim 1 of the main request involves an inventive step (Article 56 EPC 1973).

14. Furthermore, the Board considers that the remaining parts of the application as adapted meet the requirements of the EPC.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.
2. The case is remitted to the department of first instance with the order to grant a patent in the following version:
 - description: pages 1 and 2 submitted at the oral proceedings, pages 3 to 21 as originally filed;
 - claims 1 to 10 submitted at the oral proceedings;
 - figures 1 to 7 as originally filed.

The Registrar:

The Chairman:

T. Buschek

S. Steinbrener