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

Datasheet for the decision of 4 July 2008

Case Number:	T 0680/07 - 3.5.01
Application Number:	97937140.8
Publication Number:	1008042
IPC:	G06F 9/445, G06F 11/14
Language of the proceedings:	EN

Title of invention:
Fault-tolerant architecture for in-circuit programming

Applicant:
MACRONIX INTERNATIONAL CO., LTD.

Opponent:
-

Headword:
Fault-tolerant system/MACRONIX

Relevant legal provisions:
EPC Art. 123(2)

Relevant legal provisions (EPC 1973):
EPC Art. 83
EPC R. 27(1)c)

Keyword:
"Sufficiency of disclosure (no)"

Decisions cited:
G 0002/93

Catchword:
-

Europäisches Patentamt | **European Patent Office** | **Office européen des brevets**

Beschwerdekammern | Boards of Appeal | Chambres de recours

Case Number: T 0680/07 - 3.5.01

D E C I S I O N
of the Technical Board of Appeal 3.5.01
of 4 July 2008

Appellant:

MACRONIX INTERNATIONAL CO., LTD.
No. 16, Li-Hsin Road,
Science-Based Industrial Park
Hsinchu (TW)

Representative:

Horner, David Richard
D Young & Co
120 Holborn
London EC1N 2DY (GB)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted 24 October 2006 refusing European application No. 97937140.8 pursuant to Article 97(1) EPC 1973.

Composition of the Board:

Chairman: R. R. K. Zimmermann
Members: K. Bumes
G. Weiss

Summary of Facts and Submissions

I. European patent application number 97 937 140.8, published as international application WO-A-99/08186, relates to a fault-tolerant technique for in-circuit programming to update and modify sequences of instructions stored in a non-volatile memory.

II. The examining division refused the application in oral proceedings on 21 September 2006. According to the written decision dated 24 October 2006, the application did not meet the requirement of disclosure of the invention. The appellant (applicant) lodged an appeal against the decision on 19 December 2006 and paid the appeal fee on 20 December 2006. By letter dated 16 February 2007, a statement setting out the grounds of appeal and an amended set of claims were filed, claim 1 reading as follows:

"A method for providing for error recovery during in-circuit programming of a computer system including a processor (112) combined with a reprogrammable non-volatile memory (100), the method comprising the steps of:
executing an in-circuit programming handler (106) to perform an in-circuit programming process using the processor to load data or instructions into the reprogrammable, non-volatile memory;
before or during performance of the in-circuit programming process, setting (250) an in-circuit programming status (118) to an incomplete value, indicating the in-circuit programming process is in progress;

when the in-circuit programming process completes loading the data or instructions, setting (266) the in-circuit programming status to a complete value indicating the in-circuit programming process is complete; and

during initialization of the computer system, executing (216, 220, 228) a first boot code sequence (102) if the in-circuit programming status has a complete value, the first boot code sequence being programmable through the in-circuit programming process, and executing (218, 226) a second boot code sequence (107) if the in-circuit programming status has an incomplete value, the second boot code sequence being protected from the in-circuit programming process and including resources which enable the in-circuit programming handler to establish a link with a source (138, 142) of the data or instructions to be loaded into the reprogrammable, non-volatile memory and to restart the in-circuit programming process to load the data or instructions from the source into the reprogrammable, non-volatile memory, the instructions including at least instructions of the first boot code sequence."

III. In a communication dated 2 April 2008, the Board indicated, as a preliminary opinion, that the reasons given by the examining division for the refusal of the application seemed to have been essentially correct.

The appellant was then heard by the Board in oral proceedings on 4 July 2008. The matter was discussed with the appellant's representative on the basis of its requests. At the end of the oral proceedings, the Board announced the decision on the appeal.

IV. The appellant requested that the decision under appeal be set aside and a patent be granted on the basis of the claims filed by letter dated 16 February 2007 or, alternatively, if the Board could not consider the issue of inventive step, the decision under appeal be set aside and the application be remitted to the department of first instance for further prosecution.

V. According to the appellant's submissions, the purpose of the invention was to find a way how to recover from an error caused by the (main) boot code corrupted through an ICP (in-circuit programming) process, but not from an error caused by a corrupted ICP handler. The inventive idea was to store a second but minimal set of boot code, the mini-boot code, in a memory protected from the ICP process and to set an ICP status bit. During a restart with a "dirty" ICP status bit indicating a potential error situation, the system could recover and resume operation by using an uncorrupted boot code for initialising the system resources.

Obviously, if the ICP handler was itself corrupted and inoperable, the procedure would only work if the mini boot code included an ICP handler. Such a scenario was not an issue with which the present application was concerned; accordingly an embodiment with such an extended mini boot code was not described. It could not be objectionable, however, if the application did not address a problem which was possibly existing but to solve not an object of the invention.

In particular from figures 1 and 2, the person skilled in the art could easily understand how to carry out the invention. By using the ICP status value, the system was able to detect whether after a reset of the system the computer system was partway through an ICP process at the time when the reset occurred. If it was not, then the system operated as normal, if it was, then the sequence of steps 218, 224, 226, 236 were performed, prior to the ICP process resuming its normal course. These four steps ensured that the mini-boot rather than the main boot code was used to establish the resources necessary to enable the ICP handler to continue with the ICP process.

The skilled person would readily understand boot program 102 and its manner of operation. It would also understand how the resources established by the boot program enabled CPU 112 to communicate with the outside world whilst executing applications. As clearly stated at page 7, lines 11 to 15, mini-boot code 107 was an alternative set of instructions for system initialisation which was able to perform many of the normal boot functions. As stated at page 10, lines 21 to 23, when the mini-boot code was executed, it initialised minimal system resources for in-circuit programming. Further, as stated on page 9, lines 12 to 15, the mini-boot code caused the CPU to restart the ICP process by first reading a value from remote host address register 120 to determine which remote host to contact in order to reinitiate the ICP process.

The present invention was not seeking to find some new mechanism for communicating with the outside world. The mini-boot code operated in this respect in the same way as the main boot program. Furthermore, the basic operation of the ICP handler was the same as in the prior art. The person skilled in the art would readily understand the operation of the boot code, and how it assisted in providing mechanisms for communicating with the outside world, with those mechanisms then being used by the applications running on the CPU. These functions were entirely standard.

It was clear that the recovering process as shown in figures 2A to 2C was intended to download boot code and/or utility programs (see for example step 254 in Figure 2C), but was not concerned with reprogramming the ICP handler itself. Indeed, it seemed unlikely that someone would have chosen to run an ICP handler in order to re-program the handler whilst it was running.

Reasons for the Decision

1. The appeal is admissible. The appeal, however, is not allowable since the application does not meet the requirements of Article 83 EPC 1973.

2. Article 83 EPC 1973 determines that a European patent application "must disclose the invention in a manner sufficiently clear and complete for it to be carried out by a person skilled in the art".

According to the case law of the EPO, a European patent application must "contain sufficient information to allow a person skilled in the art, using his common general knowledge, to perceive the technical teaching inherent in the claimed invention and to put it into effect accordingly" (see decision G 2/93 - Hepatitis A Virus / UNITED STATES OF AMERICA II, published in OJ

EPO 1995, 275, paragraph 4 of the reasons of the decision). Sufficiency of disclosure under Article 83 EPC 1973 requires that the subject-matter claimed in a European patent application be clearly identified as from the date of filing because an insufficient identification of the subject-matter claimed cannot subsequently be cured without offending against Article 123(2) EPC (see decision G 2/93 (*supra*), paragraph 10).

The technical teaching inherent to an invention can only be perceived if the technical problem solved by the invention and the solution can be understood on the basis of the application, a requirement which also follows from Rule 27(1) c) EPC 1973 (Rule 42(1)(c) EPC).

3. In the present case, the introductory section of the application at page 2 f. (WO-publication) clearly describes the technical background to in-circuit programming of computer systems. It also identifies the object of the invention at page 3 as follows (underlining added):

> "What is needed is a method for providing fault-tolerance during in-circuit programming which can recover from an error during the in-circuit programming process, even if the <u>code used by the in-circuit programming process to communicate with the outside world</u> is improperly programmed. ...
> The present invention provides a method and an apparatus for providing fault-tolerance during in-circuit programming."

Essential communication functions of the ICP process are performed by the ICP handler as explained at page 8, line 22 ff. The ICP handler is programmable through the in-circuit programming process (see page 7, lines 4 to seven) and can thus become corrupted in the ICP process, thus causing the type of errors against which the invention seeks to provide fault-tolerance.

The appellant's arguments that recovering from a corrupted ICP handler was not a purpose of the invention are, for these reasons, not accepted by the Board.

4. To recover from an error during an ICP process, the invention proposes to run a so-called "mini-boot code 107" from a protected section of the boot code. This mini-boot code is stored in a protected memory which cannot be modified during the ICP process.

However, for recovering from an error situation, the function of the mini-boot code is merely to initialise minimal system resources and to restore the remote host address in order to reinitiate the ICP process as described at page 9, second paragraph and page 10, second paragraph. There is no clear indication how the problem of a corrupted ICP handler could be dealt with.

5. In the flow chart fig. 2A at step 240, the system initiates a link with the remote host from which the in-circuit programming code is downloaded (see page 10, last paragraph).

It must be concluded that it is the ICP handler, under whose control this step is executed. This is confirmed by fig. 2B which explicitly shows that the subsequent steps 244, 246 etc. also proceed under its control. All these steps, however, cannot be executed if the ICP handler is corrupted.

6. The appellant argued that it was obvious to include an ICP handler into the mini-boot code since if the ICP handler was itself corrupted and inoperable the procedure would only then work.

Such features of an extended set of code, however, are not derivable from the teaching of the application as filed, which is rather based on a minimal set of code, i.e. a "mini boot code ... which initialises minimal system resources for in-circuit programming" (see page 10, lines 21-23).

7. The appellant also argued that failing to recover from a corrupted ICP handler was not detrimental since the invention as disclosed allowed the computer system to recover from an error situation where the main boot code was corrupted, and this was already an improvement over the prior art justifying the grant of a patent.

The Board also rejects this argument. Although corruption of the boot code is one of the possible error sources addressed in the application, it is still the fault-tolerance against corruption of the communication code which the application discloses as object of the invention (see point 3 above). There is no hint in the application that the invention or any embodiment of the invention is restricted to be effective if a corrupted boot code is the only error source. This would be a shift of the invention resulting in a method which is not clearly identified in, nor derivable from the application as filed.

8. In summary, the Board judges that the application does not enable the skilled person to carry out the claimed invention and thus does not meet the requirement of disclosure of the invention as set out in Article 83 EPC 1973.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:	The Chairman:
T. Buschek	R. R. K. Zimmermann